UNIVERSITY OF OSLO
Department of Informatics

Federated identity management in future online markets and communities

Master thesis

Thanh Nghia Luu

May 3, 2010

Abstract

Identity overload and password fatigue is a common problem for people who engage in online activities. Federation can be a mitigation against password fatigue problem. Federated identity management aims at reducing this problem by linking multiple online identities belonging to the same party. The problem with federations are that they are best suited in closed networks, and that user-centric technologies have a greater role in open networks when it comes to mitigating password fatigue. That is why user-centric technologies will be included as a supplement to this thesis to address some of the problems with password fatigue in open networks.

The goal of this thesis is to get an overview of the available identity management solution today and have a look in some of the important aspects of federated identity management. We will then evaluate a federated system from users' perspective, and compare the experience with a slightly different way of authenticating ourselves. CardSpace and OpenID will be used together as an example to see if information cards can offer something different and convenient. We will then try to suggest future use cases for federated identity management in different areas like online services, eGovernment, enterprises and online communities. Use cases for how these areas can benefit from adapting user-centric technologies will also be included.

This thesis will then sum up some thoughts on the password fatigue problem based on the gathered information and the evaluation. It will also include some thoughts on user-centric identity as the next step in identity management.

Preface

This paper is written as part of my masters degree in computer science at Department of Informatics, University of Oslo.

I want to thank my supervisors Audun Jøsang and Pejman Bagheri for good suggestions, constructive discussions and their expertise on identity management. I want to thank my fellow students for their support and co-operation over the years. Last but not least, I would like to thank my family and friends for their encouragement, and especially my sister who has always supported me.

Nghia Luu Thanh
Oslo, May 3, 2010

Contents

List of Figures

Acronyms

SSO Single Sign-On

SLO Single Log-Out

IdP Identity Provider

SP Service Provider

OP OpenID Provider

API Application Programming Interface

SAML Security Assertion Markup Language

SPML Service Provisioning Markup Language

FIM Federated Identity Management

UCIM User-Centric Identity Management

IdM Identity Management

1 Introduction

A tendency of the internet era is the migration of sociability, business, entertainment, and other activities from the physical world to the virtual world. The Internet has become a new place of interaction between people and organizations. This new, radically different place has its own rules. Since the development of the internet had such a rapid pace, regulating digital identity had not begun until recently. Each new system, even a new application, used to come with new built-in proprietary digital identity solution. If we take a look some years back, many proprietary solutions for identity management existed, but none of the big players have success in elevating their solution to the level of wide adopted standard. Microsoft tried with their Windows Passport, but with no success as they did not get the acceptance they wanted. The need for collaboration is forcing major players towards an adoption of standards for management and interchange of identity.

One of the biggest problems today, seen from a user's perspective is the need to remember a lot of different usernames and passwords in order to get authenticated, due to the large (and growing) numbers of web services. According to a 2002 survey of British online-security consultant NTA Monitor, the typical intensive computer user has 21 accounts that require a password [18]. Since this survey was done almost a decade ago, we can only assume that this number has increased since, as a result of the increasing use of the world wide web. That means that we have to remember even more passwords. The main problem lies in our dependence on data silos. We are dealing with a lot of data silos today, which is a repository with identity information. This model is not scalable with regards to the users need on the web, resulting in reuse of the same username and password in order to make it more convenient, and consequently making it easier for malicious user to exploit their identities [28]. The user want simplicity and convenient solutions. One of the goal in this thesis is to see if federated identity management can solve some of these problems.

Federation is said to be a solution against password fatigue [8]. Single Sign-On (SSO) is one of the key ingredient in federated identity management. For Web-based Single Sign-On over decentralized identity sources, the OASIS issued the Security Assertion Markup Language (SAML) is the industry standard. Solutions based on SSO can be categorized in federated (typically SAML) or user-centric identity management (e.g., OpenID). Federated identity management is secure and most prevalent, while the user-centric approach offer better usability and maintenance.

1.1 From silos to today

Traditionally, identities were managed in so-called corporate identity silos. Silo-based identity management systems today do not scale well and the reason for keeping them is because so many users depend on them. In this model one single identity management environment is operated by a single service for a specific group of users. Hence, every (online) service had its own identity management system built on their own requirements for authorization and identification of individuals. From the perspective of users of multiple systems this means that they have to maintain an identity (account) for each and every service they use, which in practice means several sets of passwords and usernames. An obvious drawback of this scheme from the perspective of the users are that it requires them to provide the same (personal) information for every new online service. The construction of identities in these systems is guided by policies set by the provider of the service.

The next step in the world of IdM systems was the development of single sign-on (SSO). Here individuals was able to gain access to different resources (applications, web sites) within a single domain once they are authenticated. Single Sign-On was introduced on a large scale with the Kerberos protocol. Kerberos was developed at MIT and is a network authentication protocol. It is designed to provide strong authentication for client/server applications by using secret-key cryptography. The Kerberos model consists of a key distribution center (KDC), which consists of two logically separate parts: an Authentication Server (AS) and a Ticket Granting Server (TGS). Kerberos operates with "tickets" which serve to prove the identity of users. [33].

The centralized SSO model (e.g., Microsoft .Net Passport) aims to solve the silo problem, as well as lessen the burden of implementing and maintaining IdM systems within each enterprise. In this model, authentication is handled by a trusted identity provider (IdP). The IdP identifies and authenticates the user and provides a credential that can be used to access resources from associated service providers. Drawbacks of this model are that the IdP stores the user's data which creates security vulnerabilities. Furthermore, the attendance of one single IdP in all interactions on the Internet creates unwanted linkability because the IdP can trace the user after authentication. It also creates a vulnerability (single point of failure) because of the models dependence on a single IdP involved in all transactions. The business agenda behind this model is the company's ability to track the users internet use and selling this information to a third party. This information is used to either make profiles about certain users or to offer products based on the users behavior. The value of this information is very profitable.

Federated identity management (e.g., SAML, Liberty Alliance) addresses

the problems related to the dependence on a single IdP, by allowing any number of IdPs to handle authentication. The user authenticates with any of the IdPs in the federation and subsequently gain access to resources available within the federation. Some federation schemes not only handle authentication, but also allow the transfer of attributes between the participating parties (SAML). Federated identity enables entities to use the same sets of credentials, to get access to the several different autonomous services offered by all the organizations participating in the federation. Thus, making digital identities usable in different domains. Identity federation offers economic advantages, as well as convenience, to enterprises and their partners. For example, multiple corporations can share a single application, with resultant cost savings and consolidation of resources. This requires mutual trust establishment between participating organizations and some kind of authentication framework.

1.2 User-centric identity

The user-centric model of identity management has been the subject of increasing attention in recent years, as it has been identified with technologies linked to the so-called Web 2.0 (Wikis, blogs and collective services like Flickr, for example). This generally refers to web-based communities and services that facilitate collaboration and sharing between users. In fact, the user-centric model has been referred to by some of its proponents as Identity 2.0.

The user-centric model relies on the principle of giving users almost complete control on how their identity data is delivered to service providers, allowing them to use simple paradigms such as electronic cards or referring URLs. Compared to the models available today, the user-centric model is said to give the users better privacy and scalability. Two of the available solutions that are referred to as user-centric today is OpenID and CardSpace.

1.3 Usability

Windley et al. [21] states that: *"People are scared of the theoretical threats to their privacy, especially in technologies they don't understand."*

There is a point where security is not dependent on the network protocol, cryptology or system implementations. The problem lies in system interaction with the user, usability problem. The gap between a right and wrong decision today is huge. A wrong decision could mean that your computer will be infected with a spyware, keylogger or a virus. Without the proper training or understanding, a warning about agreeing to something in order to use a service can be very annoying. The result of this is somewhat

undesirable with regards to security. Although the user is warned, the benefit of not following the rules can often over weigh the risks. This leads us to one of the biggest risks in the computer world, phishing. Phishing is a product of; lack of understanding, lack of training and unusable user interfaces.

1.4 Problem description

There is a need for exchanging user credentials between several systems and applications across the organizations' boundaries. Modern users seek simple and comfortable web experience. They want the best possible and tailor-made services that fits their need. At the same time an user requires access to the resources with minimal authentication effort. Users today typically have to remember a large number of usernames and passwords, which is heavy load for users, but also burden for identity data administrators. Aside from contributing to stress, password fatigue may encourage people to adopt habits that reduce the security of their protected information. For example, an account holder might use the same password for several different accounts, deliberately choose easy to remember passwords that are vulnerable to cracking, or rely on written noted with their passwords. The ideal situation for modern users is compounded in term Single Sign On, which is basically following a simple principle: "let me just sign-in once, and let me access secured resources without being bothered". Personal data are usually sensitive, if not confidential information. Therefore the transfer of user data requires the appropriate security level assigned, to ensure data integrity and confidentiality on its way through the net. Applications with higher risk requires higher security levels to ensure confidentiality and integrity.

Federated identity management has come on board the identity ship with a mission to mitigate issues like password fatigue, administration, usability and cross-domain interaction. Federation or user-centric approach with SSO appears to be the solution for password fatigue and could greatly enhance the management within an organization. But the reality is not so easy: "Federated identity management requires a complex set of technologies and business processes, but the goal behind it is simple: to automatically share identity information across administrative boundaries" [30]. A lot of aspects regarding federated identity management has to be considered before a project of this magnitude can be realized [31]. A potential disadvantage is that the loss of a single password will prevent access to all services using the SSO system, and moreover theft or misuse of such a password presents a criminal or attacker with many targets.

Maler & Reed states that "However attractive its benefits, federated identity imposes costs as well, entailing new and increased security and privacy risks because it shares valuable information across domains using loosely

coupled network protocols." [23].

There is also another possibility when it comes to gathering your identities and managing them by using the same username and password. User-centric identity management are getting acceptance in the low-risk market, fast. The low-risk market is referring to services like blogging or social networks. OpenID is the leading technology in this field and their website[1] is saying that over 50 000 websites accepts OpenID for logins. User-centric identity is more loosely coupled and more open than federations, which make this an interesting field to explore.

We will start by get an overview of the available identity management solution today and have a look in some of the important aspects of federated identity management. The next part will be to evaluate a federated system from users' perspective, and compare the experience where we will explore a slightly different way of authenticating ourselves. CardSpace and OpenID will be used as an example to see if information cards can offer something different and convenient.I will then look at the future of federation in some areas and sum up my thoughts on the password fatigue problem.

My research questions:

- The future of federation regarding online services, enterprises, eGovernment and online communities.
- How will some of those areas benefit from adapting user-centric technologies?
- Password fatigue: How is federated and user-centric identity really dealing with this problem?

My personal motivation is to see how identity management affects users today and how the gap between the real world and internet is shrinking. Social networks are an example of how we are bringing more of our real lives into the web. That is why the authentication process has to be more seamless and connected. So that we as users can get more control over our identities and gather them in our "wallet". Information cards could be a candidate in replacing username and password, because of its intuitive metaphor and function.

1.5 Scope of this thesis

This focus of our evaluation of solutions, will be from the users perspective. Basically what the user can see and the amount of information provided from

[1] http://openid.net/get-an-openid/what-is-openid

the solutions.

The reason for including user-centric identity management (UCIM) in this thesis, although the title does not mention it, is because UCIM is in many ways the next step of identity management. Since we are going to discuss the future of federated identity, it is natural to include UCIM as well.

The problem with federations are that they are best suited in closed networks, and that user-centric technologies have a greater role in open networks when it comes to mitigating password fatigue. That is why user-centric technologies will be included as a supplement to this thesis to address some of the problems with password fatigue in open networks.

Although business intentions play a vital part of the federated identity, we will not go to deeply into it.

2 Digital identity today

Individuals are right at the center of online identity management, because it concerns the management of their identities, and because decisions are made on the basis of these identities. From an individual's point of view, the concept of identity management therefore not only relates to the access control regarding resources. It also, or maybe even rather, relates to how they are manifested and represented, and how this is aligned to their own perception of their identity. Identity management in this sense strongly relates to role playing and presentation of one self. Individuals should be able to act as autonomous individuals, be able to control their reputation, and have insight in the way they are judged by others in a specific context. The online environment facilitates the construction and maintenance of projected and imposed personas. Data can easily be collected and combined into rich personas, transcending the context in which individual bits of information were disclosed. The unrelated combination of data from different sources makes it difficult for individuals to control their different digital personas. This undermines the capabilities for people to control the image they present in different contexts and to segregate audiences online. The need to do so exists online just as it does offline. People engage in different kinds of activities online (e.g. public, commercial, and private) and need to be able to construct matching identities that meet the behavioral rules and requirements set by these different environments. Important values such as reputation, dignity, autonomy, judgment, and choice are closely related to the individual perspective on identity management. When people cannot determine or control their identity, they may become overexposed, confused, or even discriminated. Human beings have an interest in naming and sorting themselves and to play different roles. Sometimes they may even need to be anonymous and unidentified (e.g. for purposes of unpunished criticism and making mistakes). Individuals appreciate to have a diverse and autonomous life, and need to be able to adapt their identities to the environment they engage in. Even though identity management is not usually the primary goal of the individual, which may explain why many people are not eager to invest time and money in IdM systems, social values insists on the individual perspective to be taken into account in the development of identity management systems systems.

2.1 Challenges of identity management today

2.1.1 Identity and security

Security is one of the most important reasons for companies to use identity management. There can be various reasons, for instance to protect against so called hackers. But for companies the existence of hackers is not their only concern. Their own employees are one of the biggest threats. They

can access sensitive information and distribute it. Or use their computer facilities for so called unauthorized use, to use company resources for private purposes for instance.

However, motives and advantages for having secure IdM systems differs for end-user and organization.

IdM systems need to ensure that personal information that is stored in these systems cannot be obtained by unauthorized persons and organizations, for example for criminal purposes. Loss of identity information can have serious consequences for the end-user. First of all, it may lead to economic loss because some digital identities can be used to retrieve money from credit card accounts and bank accounts.

Also other adverse effects of identity misuse incur on the individual. Identities can, for instance, be abused for manipulation, deception or misuse. Inadequate security can also lead to reputational damage for the individual, for example when sensitive information becomes available to a broader public or when lost identities are being abused for criminal purposes. To repair reputation damage is difficult because it may be difficult for the "victim" to detect the behavior in real-time and because removing all damaging information may be extremely difficult in practice due to caches, backup, etc.

Identities are used as a basis of decisions and judgments, e.g. made by the government and commercial organizations like banks. Identity abuse may lead to discrimination and exclusion of services. In the worst case, this occurs without the awareness of the individual. In addition, the burden of proof of undoing discrimination or exclusion lies at the individual, which can require much effort.

Individuals also have an (indirect) interest in the security of IdM systems, because it may increase the general trust in electronic services, and thus improve the possibilities to make use of online communication and transactions. Trusted and secure IdM can pave the way for more efficient and effective services to the end-user, whereas mass loss of personal data due to insecurity will have a negative affect on the general use and supply of electronic services.

2.1.2 Identity and business

Today, you can almost buy anything on the internet. There is no need to get out of your apartment in order to buy groceries, mobile phone, books or grooming products. The business model has changed from a personal perspective over to a automated process, which also changes the trust rela-

tionship between the business and their customers.

Since most of the businesses assets are digitalized, the systems made to protect these assets has to be implemented in a way that makes the business process agile and ensure secure access. This means that whenever a user is granted access to a resource, it also has to be easy to easily deny access if necessary.

The rapid escalation of threats such as hacking, theft of electronic information, spam, viruses, and worms have clearly demonstrated the increasing need to be able to identify who is sending information and using computer resources, and to be able to check that they are acting within their authority.

2.1.3 Identity and privacy

The difference between an automated transaction and a transaction out in the physical world, is anonymity. If someone would go to a store and pay for something with cash, they would stay anonymous if the cashier did not know them. There would not be any record of them buying anything, unless they use a credit card. In an automated transaction, there is rarely anonymous transaction. You cannot pay with cash over network, making it easier to trace someones whereabouts.

Internet users are becoming increasingly concerned about what personal information they may reveal when they go online and where that information might end up. It is common to hear about companies that derive revenue from personal information collected on their web sites. Information you provide to register for a site might later be used for telemarketing or sold to another company. Seemingly anonymous information about your web-surfing habits might be merged with your personal information. Websites use cookies to gather information about users, but disabling cookies could prevent you from doing online banking or shopping at some web sites. Users has to be more involved in transactions and an emerging trend today is to require the users consent before sending personal information.

Another privacy issue is when websites sends e-mail to you to inform you that their privacy policies are changing, but most of us find it difficult and time-consuming to read and understand privacy policies or to figure out how to request that the use of our personal information be restricted. Privacy concerns are making consumers nervous about going online, but current privacy policies for web sites tend to be so long and difficult to understand that consumers rarely read them.

2.1.4 Identity and usability

The online world can be a hard place for users trying to manage their digital identities. For example, one single online session may require users to remember and fill in several different usernames and passwords. Moreover, online services can have extensive policies, requiring customers to agree on several statements before getting access to a service. For the end-user, it is often an impossible task to remember in which circumstances and under what terms they have used a service on the internet. Currently, registration or access to online services often requires obsolete amount of form filling.

Online identity management introduces many obstacles for the end-user, which may lead to very skeptical users or users that do not care (making them very vulnerable). The complexity of the online world may add difficulty for customers to actually understand why they should use an application or service.

A series of surveys, Cranor et al [10] has shown that:

- Users have difficulties using security tools correctly.
- Users do not understand the importance of data and systems for their organizations.
- Users do not believe that their assets are at risk.
- Users do not understand that their behavior puts assets at risk.

It is important to be aware of that although the users do not understand the systems, the problem is not always the user alone. The systems has to interact in a more usable way, making it easier to use and more understandable to make the right choices.

Online identity management is not what individuals care for when they interact with service providers. They engage in online transactions for the sake of transactions, the IdM is a necessary and unavoidable nuisance in many cases. Individuals often lack the interest, means, time, or knowledge to manage their identities in a way that suits their interests. To avoid that customers from engaging online interactions or make wrong decisions, it is important that IdM systems take into account the usability of the system, its default settings, and the eventual use of the system by customers. Keep in mind that usability is important, but not always the solution. The users also has to believe that by making the wrong choices, like sharing you password or writing it down, they are putting their assets at risk.

3 Research methodology

Most of my thesis is based on previous literature, mostly articles and research papers.

There is a lot of so-called ”white papers” available on the net, which means a lot of information is available, but some of them written for sale and marketing. This makes the process of sorting the truth from the overwhelming truth very hard. And the documentation of solutions seldom include very detailed information about issues and problems with their solutions, which is perfectly understandable. That is way I have evaluated some systems to find some of the issues first hand, and searched in articles and online discussion boards to get a better understanding of these problems.

4 Identity Management Concepts

Identity Management (IdM) is about how the users are identified, what rights they should have, how to control their behavior and how to organize the necessary administration. IdM consists of business policies and processes that define the goals and procedures for administrating digital identities.

4.1 Entity

The term entity refers to an organization, an user, an agent or a system. An entity is something that exists in the real world. One entity can have zero to many identities within a given domain. An example can be that a person is a student and an employee at a university.

4.2 Identity

An identity is a virtual concept and is possessed by an entity. This identity is known by a system used by the entity, and will therefore contain all the information and characteristics that the system has been able to obtain about the entity.

A system can use identifier(s) about an identity and further information, to decide what kind of access an identity has or to prove ownership of the identity. Identity is the prime concept for IdM.

Usually, the identity within an IdM system can be established with only a subset of the person's attributes. Hence, an individual user can have many identities using different sets of personal attributes for different roles and purposes. Sometimes, these identities are referred as identifiers of a user, whereas a complete identity is the union of all the user's identifiers and associated attributes.

4.3 Identifier

This is a unique index for an identity within a domain and are usually some characteristics about an identity. These characteristics may not be unique, like your birth date. Since an identifier must be unique, it can be combined by several non-unique characteristics.

"The possible characteristics of an identity may differ, depending on the type of real world entity being identified. For example, a date of birth applies to people, but not to organizations; a national company registration number applies to a company, but not to a person." [13]

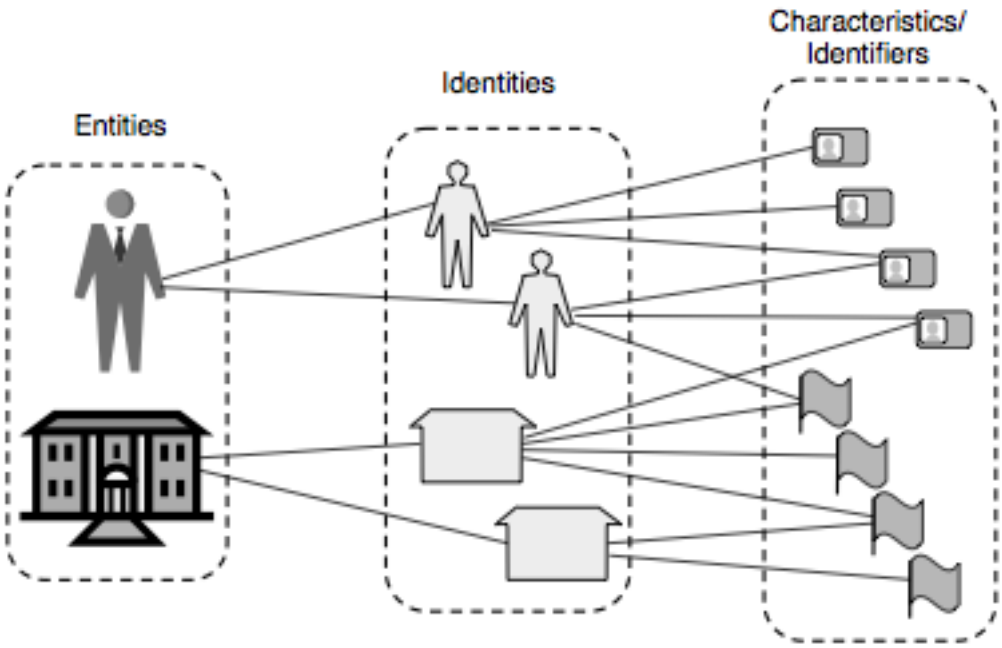

Figure 1: Correspondence between entities, identities and characteristics/identifiers. [13]

An identifier can be transient or permanent, self-defined or issued by an authority, suitable for interpretation by humans and/or computers.

4.4 Attributes

Once the identity is established, a wide variety of attributes and objects might be associated to it. Some of these associations might be formal, specific relationships amongst peer objects (e.g., people and bank accounts). Others might be informal, loosely-linked affiliations (often one-to-many and many-to-many) which may change frequently over time. Another link would be people with role-based rules. An example might be that a person as an engineer might have a certain level of signature authority. When that person steps in as acting manager when the manager is on vacation, there will need to be an alternative authority which may be the engineer. Thus, temporary privilege changes could be supported by acknowledging a time window for the additional role.

Attributes associating with the identity have different perspectives from different parties. And yet, the identity can be dynamic as these attributes like the user's reputations can change over time.

4.5 Digital identity

Digital identity is a part of overall identity - identity information translated in bits and bytes. The identity data need to be created, stored, exchanged over electronic networks, used, copied, deleted, retrieved, manipulated etc.

Digital identity usually describe user's physical identity in electronic formats for computer applications. Therefore, various aspects of the person's attributes can be used to represent the person in identity management. These attributes can include the name of the user, biological characteristics and other properties such as the person's reputations and affiliations.

4.6 Authentication

Authentication is the process of verifying somebody's or something claim of holding an identity. Your identity may be proven in a number of ways e.g. fingerprint, signature, or simply by showing your ID card. Looking at an individual isolated from the rest of the society, identity is something one **IS**, something one **DO**, something one **KNOW**, and something one **HAVE**. In private and public computer networks (including the Internet), authentication is commonly done through the use of regular passwords. Knowledge of the password is assumed to guarantee that the user is authentic. Each user registers initially (or is registered by someone else), using an assigned or self-declared password. On each subsequent use, the user must know and use the previously declared password.

This requires a process for authentication and an authentication authority. Generally, the identity issuer tends to be the authentication authority. When the only requirement of the identity is uniqueness from other identities, the process of authentication may be quite lax. As the requirements become more strict, the process evolves from a simple password to two-factor validation and beyond.

Authentication handles three responsibilities regarding identities:

1. Registration
2. Credentials management
3. Entity authentication

4.6.1 Username and password

This is the classic log-on mechanism and has been around for many years. An administrator issues people with individual identifiers and an initial password. The user logs on and has to change the password to something new to ensure that it is a secret kept even from the administrator. Sometimes there are rules about how long the password is, letter or number combinations or special characters, how often it has to change, etc. It is cheap to implement and easy to administer. It can be used to enable cryptographic services. It is open to being stolen by many methods, and systems that do not detect too many attempts to use the wrong password are open to computerized attack. Notice that a password does not authenticate a person, successful authentication only implies that the user knows a particular secret. There is no way of telling the difference between the legitimate user and an intruder who has obtained that user's password.

4.7 Authorization

Authorization is when a person or an operational entity has gained the required authority or permissions to do an operation or a task. Authorization is mostly proceeded by authentication. Once the user is authenticated, the authorization process checks what rights the user has.

In computer systems, authorization is where the system administrator or similar authority translates a user's (or a specific group or class of users) permissions to access a set of system resources, data files, programs, specific functions and commands, networked facilities, etc. into computer-recognized form for binding to that user's authenticated identity.

The system can determine the kind of authorization based on either the user's identity or the method used to authenticate the user.

4.8 Access control

Access control refers to the control mechanisms that ensure access and permissions are given to all those who have the required access rights (an authenticated user with the required bindings) to perform specific operations on the resources within that system. This same mechanism denies access to unauthenticated users and to authenticate users if they attempt to perform any operation that they are not authorized to perform.

Systems relying upon access controls usually have lists of the users who are allowed to access the various resources available within it, together with the rights that they have. These lists are referred to as access control lists.

Access control is more or less the implementation of the security policy. The policy describes which rights an authenticated person has.

4.9 Identity management

Identity management is about handling identities throughout its life cycle and its main processes are authentication and authorization.

Identification is the process of determining a subject's identity. Process of determination is also called authentication process. During the authentication process a subject proves her identity indicated by presenting some credentials. E.g., if a user logs on to the IT system, she indicates an identity using a user name and proves her identity by entering valid password.

The identity management process starts with an entity claiming an identity. Given that the requirements are met, the entity is given an unique identifier (username) within the system.

The identifier itself is not sufficient proof to be authenticated. In order for an entity to be authenticated and proof its claimed identity, is to provide the system with e.g. a password. This information is only known to the entity.

Once the entity has been authenticated, as far as the system knows, the entity is the rightful owner of the claimed identity. At this point, the authorization process starts. The system makes sure that the identity is only allowed to perform the actions it has the permission to do. These permissions are set by the administrators and nothing the user can customize after its needs. Access control mechanism takes over, and makes sure that the rightful owners with the right permissions has access to a resource.

4.9.1 Identity life cycle

In a typical identity management system the life cycle of an identity consists of three stages:

- Enrollment
- Maintenance
- Disposal

In the first stage of the identity life cycle, users register with the organization using their personal attributes and credentials. Once the information provided by the users had been verified, the IdM system will add the users

and create unique identifiers for them in the directory. After the enrollment, the identities will be provisioned with access controls as well as services entitled to the users. These identities will be maintained in the directory for later stages of the identity life cycle.

During the operational life of the identities, users can read, modify and delete their identity information stored in the directory. In the meantime, the IdM system maintains the information and audits or reviews the operations to ensure that the changes made by the users are valid. It can also suspend and resume identities from accessing to services. Moreover, the identity information may be shared with other applications.

Finally, the last stage of identity life cycle is the disposal of identities. In this stage, the identities of the users will be deleted and removed from the directory. And consequently, the services entitled and associated rights will also need to be revoked. After this process, generally the identities and their information will be archived.

4.10 Trust

Trust is something we all understand at a human level, but not necessarily when it comes to business-to-business relationships or to the technical systems needed to support business relationships.

The dictionary describe trust as:

> *Trust: Firm belief in reliability, honesty, veracity, justice, good faith, in the intent of another party to conduct a deal, transaction, pledge, contract, etc. in accordance with agreed principles, rules, laws, expectations, undertakings, etc.*

So basically, trust is a subject's willingness to believe the claims asserted by another subject. In identity-based transactions there are usually three roles involved: subject, relying party and identity provider.

A subject is something or somebody who owns a digital identity, e.g. a user. A relying party is an entity that provides a service intended for a restricted audience, e.g. a web shop. An identity provider issues digital identities, e.g. your bank.

Patrick, Briggs and Marsh [10] divides trust into three layers:

- **Dispositional trust:** A persons natural instincts to trust someone or not.

- **Learned trust:** A person's general tendency to trust, or not to trust, as a result of experience.
- **Situational trust:** How a person uses signals, like the amount of information given or social expectation, to decide whether to trust someone.

For example, if a person is in a unknown situation she would rely on "dispositional trust" rather than "learned trust". But if a experienced computer expert where to shop online, she would be able to rely on "learned trust" in order to decide whether the web shop is safe or not.

The ability to perform a given action depends on user's level of knowledge. A user would generally consider the competence of a service, before letting them perform a given task. For example, you would not give your car to a garage if you did not think the mechanics could repair the car. Trust is given then only when the user has faith in the competence of the other party. Signals that influence this process can be reputation, past experiences, or certificates confirming the competence of a service.

Most people have long experience in judging the many different situations in relation to confidence in your daily life outside the internet world. Some users their trust solely on the information provided on the website, while others are highly skeptical of any kind of internet based services. Users will use their perception to evaluate a site based on its off-line reputation (if possible), information and reviews from other users. However, their willingness to only accept the information on a website could change when they are dealing with services of higher risk (online banking or eCommerce). But the common denominator whether the service is of high risk or not, is the website's ability to offer a good user experience in addition to security indicators (like security seals) [5].

4.10.1 Risk

Risk is based on assessing what the loss might be if something goes wrong, and whether you can absorb that loss if it does go wrong. Thus, we have levels of trust. For a small-value transaction, the degree of confidence in a trust assessment does not have to be large; for a multi-million dollar transaction, the level of trust needs to be very high. The required level of trust depends on your business policies on trust and risk management. A very common risk management approach is staged payments and bank bonds; another is to cover unacceptable financial risk by insurance.

Technology-dependent businesses need to enable appropriate risk decisions to be made. Trust services can be provided to automate steps in the

business process to build trust, checking identity credentials on people and institutions, authenticating sources, etc.

There has been a tendency within the IT community to misrepresent "trust" as a single process at a point in time, whereas trust is a process in itself. Trust is built or destroyed over time. Trust is generally subjective, though it may be supported by empirical information. This has resulted in the user community losing confidence in IT solutions providing a reliable basis for trust.

The delivery of empirical information in support of trust services by electronic means is both practical and necessary for the future growth of e-business. This development should enable more improvements in business process efficiency and better control over the business transactions.

4.10.2 Design

Fogg et al [6] conducted a survey at Stanford University Persuasive Technology Lab at the request of the Consumer Web Watch in 2002. In survey comments from 2440 participants categorized into categories illustrated in Figure 2.

The survey shows that 46.1 % of users had design and appearance as a important factor in the evaluation of the site's credibility. 28.5 % mentioned information design / structure in their comments, while 25.1 % relied on information focus.

4.11 Privacy

People want their identity data protected, and they want to preserve their privacy. Some companies are very dependent on privacy and need to appoint privacy officers to ensure the privacy of their customers and themselves.

Many examples show that people are willing to exchange their personal data for some payoff they understand. Companies are concerned about privacy because they want to keep their customers: if the customers' needs are not met, they tend to switch supplier. Privacy policy is often part of terms of service agreement. Some companies conducting e-business use cookies to recognize the customers next time they return, to track their purchasing habits or their behavior on the net. Created profiles are usually used for direct marketing, but they can also be subject to misuse in form of selling them to third party.

Customers have the right to know which data are collected, why, and

	Percent (of 2,440 comments)	Comment Topics (addressing specific credibility issue)
1.	46.1%	Design Look
2.	28.5%	Information Design/Structure
3.	25.1%	Information Focus
4.	15.5%	Company Motive
5.	14.8%	Information Usefulness
6.	14.3%	Information Accuracy
7.	14.1%	Name Recognition and Reputation
8.	13.8%	Advertising
9.	11.6%	Information Bias
10.	9.0%	Writing Tone
11.	8.8%	Identity of Site Operator
12.	8.6%	Site Functionality
13.	6.4%	Customer Service
14.	4.6%	Past Experience with Site
15.	3.7%	Information Clarity
16.	3.6%	Performance on Test by User
17.	3.6%	Readability
18.	3.4%	Affiliations
(Categories with less than 3% incidence are not in this table.)		

Figure 2: How often participants commented on various issues when evaluating the credibility of Web sites. [6]

what the companies do with it, but not many get to know it. It is important that companies understand the level of identity needed for successful business interaction. Mitigating customer expectations is usually a guaranty for long- term relationship.

How is so privacy connected to DI and common IM infrastructure? Individuals will use IM only in case they feel secure on the net, but they will ignore it if they feel they are under constant surveillance. In order to be able to protect identity data, and preserve privacy, chief privacy officers have to know all about the data from the moment of their creation to the moment of their destruction. Sooner or later it is necessary to make data inventories or resource mapping, and that is the first step in performing privacy audit. To complete this audit many other issues have to be addressed such as data collection, ownership, custodianship, access, storage, transport, backups, logs and other security measures, but first of all control over identity life cycle have to be preserved.

4.12 Digital Certificates and PKI

A private/public key pair allows signing and encryption of messages like e-mails and can be used for authentication. The possession of a key pair does not provide a way of verifying that the user is who he pretends to be. To complement this information someone is needed who asserts that the owner of a specific key pair is identity X. That someone is the CA and the attestation used is the certificate. A certificate contains information that identifies the certificate's owner (called the subject) as an entity on the network. A certificate also contains the owner's public key and a validity period of the certificate. Furthermore, a certificate identifies the CA (called the issuer) that issued the certificate. The information is signed by the CA, so that information cannot be altered. This kind of certificate is also known as X.509.

A certificate is similar to your passport, which states that you are Mr. or Mrs. X and has some sophisticated methods to assure that the identity statement cannot be modified.

Before public key cryptography can be widely used and easily managed on public networks, a public key infrastructure must be in place. Without a public key infrastructure, public key technology is not generally suitable for large-scale enterprise deployment. A public key infrastructure (PKI) provides the framework of services, technology, protocols, and standards that enable you to deploy and manage a strong and scalable information security system based on public key technology. The basic components of a public key infrastructure include digital certificates, certification authorities, registration authority, validation authority and some kind of certificate revocation

list (CRL).

The Registration Authority in the real life example mentioned above would be the passport authority and the binding would be done through a certificate of birth, a social number or similar, depending on the country you are in.

An example of a real life Validation Authority is the passport inspection when entering a country. They check your passport by comparing the photo or other biometric methods in your passport with the person itself, which means s.th. noted in the passport will be compared to something the person has.

In the internet world such a visual inspection might not be suitable. In that case this "something a person has" is his private key. By encrypting or signing a message the sending person is identified because it is only the matching public key that is able to decrypt or verify the message. Yes, exactly that public key wrapped inside the certificate which states the binding of Mr. or Mrs. X to the public key.

In order to check the certificate's validity we have a CRL. The CRL is a list of subscribers paired with digital certificate status. The list enumerates revoked certificates along with the reason(s) for revocation. The dates of certificate issue, and the entities that issued them, are also included. In addition, each list contains a proposed date for the next release. When a potential user attempts to access a server, the server allows or denies access based on the CRL entry for that particular user.

4.13 Single Sign-On

Single sign-on (SSO) is mechanism whereby a single action of user authentication and authorization can give a user to access all the services where he has access permission within a environment, without the need to enter a password multiple times. Single sign-on reduces human error and increases user experience.

Single sign-on (SSO) is a form of access control, with SSO users can authenticate once and get access to multiple systems. SSO has many applications, it is not only used within companies to allow users to login once and then use all applications that they need without having to re-authenticate. People can also use it to access web sites, applications and other systems. With programs like OpenID and Windows Cardspace you can store multiple identities for the various applications you have. In those programs your identity including your password is stored. When you start your computer you

authenticate to one of those programs and the program then authenticates you if you try to access a resource.

As single sign-on provides access to many resources once the user is initially authenticated (”keys to the castle”), it increases the negative impact in case the credentials are available to other persons and misused. Therefore, single sign-on requires an increased focus on the protection of the user credentials, and should ideally be combined with strong authentication methods like smart cards and one-time password tokens.

Single sign-on also makes the authentication systems highly critical, their failure or an inability to reach them (such as in a network failure) can result in denying access to all systems unified under the SSO. This can make SSO undesirable for systems to which access has to be guaranteed at all times. The problem can be solved with replication of IdPs.

4.13.1 Single Log-Out

Single Log-Out (SLO) enables users to cleanly close all their sessions in a SAML landscape, even across domains. Not only does this save system resources that would otherwise remain reserved until the sessions time out, and SLO also mitigates the risk of the hijacking of unattended sessions.

SSO means the user logs in once and has access to multiple applications. SLO means that the user can sign out once and will be logged out of all applications he accessed during the federated session. Single sign-out is explicitŮthat is, the user has the intent to sign out; expired session cookies or tokens do not result in single sign-out.

5 Identity Management Models

IdM can be divided into server-centric and user-centric systems. A server-centric IdM system is a centralized system for managing accounts, managing user data and associating data to subjects in a convenient way.

The users in a server-centric IdM system rely on the identity provider to present credentials to others. In this case, the users have little or no control over their credentials [4].

In a user-centric IdM system, where the system is an user-oriented system, the users are able to have more control over their credentials. The users have the capabilities to decide on what identity information to be disclosed. They become less dependent on their identity providers, since the long term credentials can be obtained and stored under their control [4]. Recent user-centric IdM has been focusing on privacy protection, which aims to minimize the exposure of the users attributes. This is to protect their identities from being misused by malicious users.

IdM systems can also be categorized into four models: silo, centralized, federated and user-centric IdM model.

5.1 Silo

The silo model is an isolated IdM system, where each system manages the identities of the users and related information in its own domain. This means that each system consists of only one identity provider and one service provider. Normally, the service provider in such model can also act as an identity provider for authenticating users and managing tokens [21].

The silo model is easy to implement and it provides tight controls over identities as only one entity, i.e. the identity provider, is exposed to the information [21]. But, it is inconvenient for users who wish to obtain access to services from multiple service providers. This is because each service provider has its own set of rules and processes. Therefore, the users will be required to register at different service providers for different services.

5.2 Centralized

Another type of model is the centralized IdM, with only one identity provider in charge of identities and authentication for several service providers. This simplifies the control procedures within the service providers. The user experience is improved by offering the SSO functionality, which is not the

case with the silo based model. However, the involved service providers will only need to trust one identity provider, making the identity provider the single point of failure. Josang et al. [21] talks about three variants of the centralized IdM.

5.2.1 The Common Identity Domain

has a central authority. This authority takes the role of the identity provider for managing identities and tokens, but it does not handle authentication during service access. Generally, PKI is implemented for this common domain, it manages and issues public-key certificates as authentication tokens.

Simple management for service providers give this model an advantage. As well as easier management for the users as they only need to obtain one unique identifier and authentication token from the common identity domain. Despite the advantages, the privacy of the users may be comprised since the SPs may be able to match identity information using the common identifier. Moreover, it is difficult to implement this kind of model, especially when it comes to defining unique identifiers for users from different regions.

5.2.2 The Centralised SSO

includes the SSO functionality for authenticating users. This model sends security assertions directly or indirectly to service providers once the users have been authenticated. Google Apps [17], is an example of this type of centralized model. This model has similar advantages of the centralized IdM model mentioned before, where users have a more convenient way to access services. However, the model also suffers from the above disadvantages.

This model is more suitable for closed environments rather than open environments, because of its simplicity in implementation of a central identity provider. Especially if the service and identity providers is under the same organization and has the same authentication policies. According to Josang et al. [21], Kerberos Authentication and Active Directory can be useful for implementing this model in closed networks.

5.2.3 The Centralised Model with Browser Support

is similar to the centralized SSO identity model, but with an additional browser support, Windows CardSpace or InfoCard. The Windows CardSpace is the client component by Microsoft, which acts as a broker between the identity providers and service providers. Whenever the users request for a service, they will first be asked to select an identity stored in CardSpace. Next, the CardSpace will communicate with the identity

providers that handles the attributes belonging to the user. The identity providers will return a security assertions to CardSpace, which then forwards these to the service providers in order to access the service. This type of model aims to improve the user experience and avoid single point failure, with the help of multiple identity providers.

Moreover, this model as well as the other two centralized IdM models only supports users within one identity domain. Hence, if users wish to access services from another domain, they will need to obtain new sets of identifiers and authentication tokens from identity providers of that particular domain.

5.3 Federated

has the functionality of identity federation, which serve to enable the portability of identity information across otherwise autonomous security domains. The model in general consists of many independent IdM systems or silo domains, where each system has its own service provider and identity provider.

The participating organizations in a identity federation has to mutually agree on; a set of policies, standards and practices. This makes it possible for the participants to collaborate with identity and service providers across different domains. For better user experience, identity federation can apply SSO functionality. Like the centralized SSO identity model, an identity provider gives a security assertion about a certain authenticated user to the service provider within the identity federation. The service provider will then, based on a pre-established trust relationship, accept the assertion and allowing the user to gain access. One major difference between the centralized model versus the federated model, is that the latter one is able implement SSO functionality in a open environment. A challenge in this model is to establish trust between the different parties. Moreover, the privacy of the users may be compromised as identifiers in different service providers are mapped. The model, as well as the centralized models and the silo model, suffers from scalability and password fatigue problems as the SSO functionality only enables users to communicate across multiple domains within a federation [2, 21].

Identity federation is a broad concept. It can relate to systems with both a high level of security as to systems with a low level of security. Moreover, federated IdM systems come in the form of user-controlled systems, but are usually controlled by businesses or governments. In addition, FIM-systems can be ”token-based” or ”anonymous-credential-based”, meaning that some systems rely on the mediation of Identity Providers (IdP) between Service Providers (SP) and the user, whereas other systems let the user construct her identity out of anonymous credentials.

However, FIM-systems have in common that agreements, policies, and standards are used to make identities portable. Also, they often rely on schemes of Single-Sign-On (SSO), even though credentials and identities may be stored at different locations, under different conditions.

Combinations of the following features are provided by federated identity management:

- **Identity provisioning:** Based on the registration to one service, respectively identity provider, several services providers are able to generate accounts for that particular user and based on this account to authenticate her.
- **Single-Sign On:** Based on a login to one service, respectively identity provider, the user is also able to user her existing accounts for other service providers.
- **Attribute exchange:** The linkage of several attributes of the user to one digital identity in the domain of one service, respectively identity provider, could be requested from other service providers as well, at least under certain conditions. The exchange of attributes also facilitates authorization. From the user's point of view federated IdM makes it more convenient for the user to gain access to several services. It relieves her from the burden of remembering and utilizing several account names and passwords, for example. Furthermore, FIM can be beneficial for professional organizations, because data storage and identity distribution can be made more efficient.

Moreover, FIM makes it possible to outsource storage of data and provision of identities. Besides these advantages of federated IdM there exist also challenges with regard to privacy and security issues for users personal data as well as the security goals of the organizations taking part in the federation.

5.4 User-centric identity

One of the main differences between Identity Federations and User-Centric Identity Management lays in the fact that a user is not necessarily associated with a home institution, which handles the user's data. On the contrary the user can gather the data requested by a resource, assemble them into a token or an URL and deliver this to the resource the user wishes to access. This allows users to interact flexibly with multiple services. Two main solutions have been proposed with respect to user-centric identity management:

CardSpace from Microsoft and OpenID managed by the Open ID Foundation. Both organizations have announced a strategic collaboration to make both solutions compatible in future.

Although detection of identity theft and misuse is not the primary focus of these identity management systems, well-designed and user-centric identity management systems are expected to reduce such risks.

6 Federated Identity Management System

We will have a look at some of the fundamental parts of federated identity management.

6.1 Identity provider

IdP (Identity Provider), is a system that creates, maintains, and manages identity information for entities and provides authentication to service providers within a federation or distributed network. It is a trusted third party that can be relied upon a transaction when users and servers are establishing a dialog that must be authenticated. The IdP sends attribute(s) containing trusted information about the user to SPs.

6.2 Service provider

SP (Service Provider, sometimes called the relying party) validates the users in a transaction and offers them a service or services. The service provider is responsible for controlling access to services, validate an asserted identity from an IdP and provide access based on the identity and manage only service related user information.

In some scenarios, an organization might act as an IdP and SP. For example, where government agencies access each other's applications, each agency plays the role of identity provider or service provider, based on the context of use.

6.3 Trust

Identity federation allows both types of organizations to define a trust relationship whereby the SP provides access to users from the IdP. This trust relationship is critical for the setup and success of a federation.

Trust is very important whenever a federation is about to form. The technology and standards has to of course be in place before parties can form a federation. The SP for example needs to trust the IdP to properly authenticate the user, possibly even authorizing the user for specific services at the SP. Without trust this scenario cannot be realized. The trust agreement and rules should be defined in a way that every party in the federation can trust each other. If party A trust party B, and part B trust party C. Party A should be able to trust party C as well. These three parties form a "circle of trust", which consist of a set of agreement and rules, for sharing information and cooperation. Each party is responsible for the security of information flow and that their systems are up-to-date, since the security

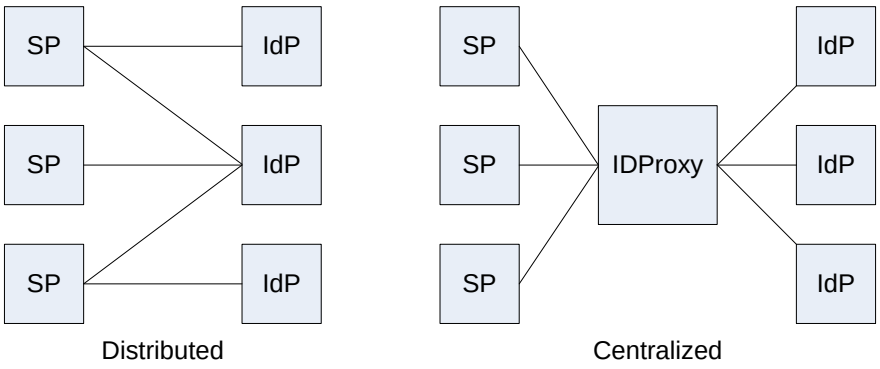

Figure 3: Distributed and centralized topology.

chain is not stronger than its weakest link.

The different parties has to agree to:

- trust in each others claims
- trust that the parties maintain their responsibilities
- the same rules in terms of security and assurance

6.4 Topology

6.4.1 Centralized

In this topology, SPs and IdPs are linked through a proxy. In e.g. FEIDE, which is centralized, the proxy sends the users to their IdPs based on their affiliation. There is a trust relationship between proxy and IdPs, and proxy and SPs. This means that the SPs trust all the IdPs connected to the proxy. FEIDE[2] in Norway and WAYF[3] in Denmark uses this topology.

The beneficial sides of this architecture is better maintenance and simplicity. The IdPs can instantly get access to all the SPs by just connecting to

[2]http://www.feide.no
[3]http://wayf.dk

the proxy. New services can be easily introduced and the proxy can detect problems with either SPs or IdPs (central point for problem detection). In terms of implementation cost, there is only one integration point compared to the distributed model. The number of connections inside the federation is minimal and the proxy can support several technologies like Shibboleth or SAML (easier to connect regardless of technology).

Some of the problems regarding this architecture are single point of failure and the amount of responsibility and trust given to the proxy. If the proxy goes down, the whole system goes down. Since every login is done through the proxy, the proxy can be a bottleneck if the necessary performance hurdles is taken care of.

6.4.2 Distributed

A distributed model is based on the idea that the IdP has to connect to SPs themselves as long as they both belong to the same federation. This topology is used by e.g. Haka[4] in Finland and SWAMID[5] in Sweden. So some kind of trust relationship has to be established before they can communicate.

This topology does not suffer from the single point of failure problem introduced in the centralized model.

One downside with this model is that it is hard to discover errors in the federation. Another issue is that the IdP or the SP has to implement a new connection every time they want to introduce a new service. This could slow down the adoption of new services.

There is a version of the distributed model, where the IdPs and SPs does not have a pre-established trust relationship. An example of this version is OpenID. The user provides her IdP through their URI, which the SP process and sends the user to the IdP for authentication. The SP and IdP communicate with each other to ensure the integrity of the claims from the IdP. In other words, that no one has tampered with the assertion.

Although, in theory, a circle of trust can be established so that a SP only trust a certain number of IdPs. This is similar to the e-government project proposed by United States [15]. In order to use one of governments AAL-1 services, the users has to use one of the government trusted OpenID-providers.

[4]http://www.csc.fi/english/institutions/haka
[5]http://www.swamid.se/

6.4.3 IdP discovery

To provide portal-style IdP-initiated SSO, administrators can typically configure an IdP to contact its partner SP sites when Alice wants to visit them. But with SP-initiated (bookmark-enabled) SSO, we encounter the IdP discovery problem that is, how does an SP know where to send its authentication request when Alice visits and wants an identity-based service? This "where are you from?" problem has a few possible solutions.

If the SP is in a prearranged IdP partnership (a "circle of trust" that often involves business contracts and legal liability agreements), we can statically configure it with the IdP's location. If the SP must choose from multiple IdPs that is, if it has no established IdP relationships, its circle of trust includes multiple IdPs, or it belongs to several circles of trust. Alice might have to input her IdP's location. This scenario is known as simplified (rather than single) sign-on. Here, the process is not seamless, which exacts a significant cost when attention and usability are at a premium. Another option is to give Alice a user agent that's smart enough to know the answer. As Web browser limitations become more risky and devices such as smart phones gain popularity, the role of "smart clients" (e.g. PAD) is becoming increasingly important.

6.5 Interoperability Standards

All federated identity models are based on interoperable standards and existence of an identity management infrastructure. Interoperability includes defining and adopting standard languages and protocols for interchange of identity data. OASIS, the Organization for the Advancement of Structured Information Standards is the international organization for standardization that has come furthest with the development and application of open-source standards. SAML and SPML are OASIS standards. The OASIS standards enable interoperability between identity systems.

6.5.1 Security Assertion Markup Language (SAML)

SAML is like "an envelope" for interchanging identity information. SAML defines an XML-based framework for transporting security and identity (e.g. authentication and attributes) information between computing entities. SAML promotes interoperability between heterogeneous security systems, providing the framework for secure e-business transactions across company boundaries. By abstracting away from the particulars of different security infrastructures (e.g. PKI, Kerberos, LDAP, etc), SAML makes possible the dynamic integration necessary in today's constantly changing business environments. SAML is a product of the OASIS Security Services Technical

Committee.

SAML does not standardize all aspects of identity management. SAML addresses one key aspect of identity management, how identity information can be transported from one domain to another. A typical alround identity management solution would also define mechanisms for provisioning, authentication or access control.

SAML divides the authentication and authorization in two parties (similar to Kerberos): the Service Provider (SP), being the party that holds the resources the user wants to access, and the Identity Provider (IdP), that holds the identities of the users for example at a local institute (called "home organization"). SPs and IdPs are joined in a federation. Within the federation all SPs and IdPs trust each other. The trust is established using X.509 certificates. These certificates and addresses to resolve the handlers of the SPs and IdPs are stored in a metadata file that is distributed to all participants and builds the federation. Identity Providers issue a digitally signed assertion (or token) authenticating the user. SPs use the certificates of the IdPs published in the federation to verify the digital signature and accept the authentication. Authorization is done using attributes that the IdPs sends along with the assertion.

SAML is a flexible and extensible standard designed to be used - and customized if necessary - by other standards. In practical terms, SAML consists of a set of specifications and XML schema, which together define how to construct, exchange, consume, interpret, and extend security assertions for a variety of purposes.

The SAML standard in the current version 2.0 guarantees interoperability between SAML implementations of different software manufacturers (e.g., Shibboleth 2.0).

The main features of SAML 2.0 are as follows:

SSO: SAML provides a standard for cross-domain Single Sign-On (SSO). Other methods exist for enabling cross domain SSO, but they require proprietary solutions to pass authentication information across domains. SAML 2.0 supports identity-provider-initiated SSO as in SAML 1.x. SAML 2.0 also supports service-provider-initiated SSO.

SLO: Single Log-Out (SLO) enables users to cleanly close all their sessions in a SAML landscape, even across domains. Not only does this save system resources that would otherwise remain reserved until the sessions time out, but SLO also mitigates the risk of the hijacking of unattended sessions.

Identity federation: Identity federation provides the means to share

identity information between partners. To share information about a user, partners must be able to identify the user, even though they may use different identifiers for the same user. The SAML 2.0 standard defines the name identifier (name ID) as the means to establish a common identifier. Once the name ID has been established, the user is said to have a federated identity.

6.5.2 Service Provisioning Markup Language (SPML)

An XML-based framework, being developed by OASIS, for exchanging user, resource and service provisioning information between cooperating organizations.

The goal is to allow organizations to securely and quickly set up user interfaces for Web services and applications, by letting enterprise platforms such as Web portals, application servers, and service centers generate provisioning requests within and across organizations. This can lead to automation of user or system access and entitlement rights to electronic services across diverse IT infrastructures, so that customers are not locked into proprietary solutions.

For example, a supply partner (Company A) goes to its partner's (Company B) supply chain portal and requests access to its inventory data, which is stored in a back-office system. In response, Company B initiates a request using SPML to communicate with SPML-enabled identity management software. After automatically acquiring the appropriate permissions, Company B grants the appropriate access levels to Company A to gain access to the data it needs.

This process takes place without the need for the portal environment to have an intimate understanding of the back-office environment. In other words, it's all automatic. The prototype encompasses all of the provisions of the proposed SPML standard while also leveraging the benefits of the Security Assertion Markup Language (SAML).

6.6 Authentication Frameworks

The participating organizations in a identity federation has to mutually agree on; a set of policies, standards and practices. This makes it possible for the participants to collaborate with identity and service providers across different domains. A authentication framework is supposed to:

- facilitate improved interoperability across the different domains by establishing a consistent approach to authentication within the federation.

- set the requirements for joining the federation
- define assurance levels for authentication

Authentication framework consists of registration and authentication. The goal of such a framework is to determine the authentication requirements and risks, and suggest the most appropriate assurance levels for authentication.

When multiple service providers are using the one authentication mechanism infrastructure, each service provider must only access the identity information they have collected, and the ability to access other client information not collected by the service provider must be prohibited unless the client's permission to do so has been gained.

Authentication assurance levels

Different types of services require different levels of authentication assurance. For example, services involving sensitive information or financial transactions would require a higher level of assurance about the identity of a client than services which do not. It is important to provide a level of authentication assurance that is appropriate for the service. This is necessary for a proper service and in order to prevent improper use and fraud. It is also necessary to ensure that risks of the participating parties are managed and clients are protected.

Authentication Assurance Levels indicates the level of certainty the system has that an user is in fact the user that the system has registered. In other words, that the user John Smith is behind the computer and not his mother logging in on his behalf. For example, John Smith can authenticate himself with his username and password to prove his identity at level 2, but has to meet up personally to prove his identity at level 4 (which is the highest assurance level in e.g. the Australian e-framework).

6.6.1 Authentication Framework in Australia

The National e-Authentication Framework (NeAF) compiled by the Australian Department of Finance and Deregulation, is an extensive framework aiming to assist agencies, jurisdictions and sectors in authenticating the identity of the other party to a desired level of assurance or confidence [25]. This framework is said to be one of the best and most extensive frameworks in the world. The framework has gotten its inspiration from USA, UK and several other countries frameworks.

No assurance	Minimal assurance	Low assurance	Moderate assurance	High assurance
Level 0	Level 1	Level 2	Level 3	Level 4
No confidence is required in the identity assertion.	Minimal confidence is required in the identity assertion.	Low confidence is required in the identity assertion.	Moderate confidence is required in the identity assertion.	High confidence is required in the identity assertion.

Figure 4: NeAF Authentication Assurance Levels [26]

The NeAF includes a methodology for becoming a member of the eGovernment federation:

The NeAF methodology steps are [25]:

1. *determine the business requirements*
2. *determine the assurance level requirements*
3. *select the registration approach*
4. *select the e-Authentication mechanism*
5. *select an implementation model*
6. *assess the business case and feasibility of the implementation model*
7. *review the e-Authentication solution.*

These steps are just a part of an even wider information security risk management process required by the agencies wanting to participate. Aspects like Roles and Responsibilities, and Authentication of government websites (server authentication) has to be addressed as well.

Figure 4 shows us that the NeAF assurance levels. To determine an appropriate overall Authentication Assurance Level for services, the Australian NeAF have to include two separate processes in an authentication process (registration and subsequent authentication). The overall Authentication Assurance Level (AAL) achieved for a service is dependent on both the registration process and the subsequent authentication process which occurs during each service request. That is, the assurance that can be held in these two processes combine to provide an overall authentication assurance level.

This citation from the NeAF documentation [25], illustrate the importance of assurance levels:

Central to the NeAF is the concept of assurance levels. An assurance level is determined through a comprehensive risk assessment process that determines the severity of the impact of getting e-Authentication wrong. While the NeAF notes that e-Authentication is one of the possible risk mitigation solutions that can be adopted to address identity-related risks its focus is on answering the questions "Do we have the correct party at the other end of the line?" and "Are they who they purport to be?"

6.6.2 Authentication Framework in Norway

Framework for authentication and non-repudiation created by Ministry of Government Administration Affairs (FAD) [24] is an aid for parties that want to interact and share data with public services. The framework is supposed to be technology independent so that the framework can be used throughout the public sector. The goal is to reuse the same authentication service across the public sector and make sure that every party adapt to the purposed assurance levels.

The framework is just an illustrative guide, it is up to individual agencies to secure their services and perform the assessments that are needed.

Four assurance levels are defined in the framework based on consequences of security breaches. The larger the consequence are of a security breach the higher assurance level will needed [24]. High risk application is related to assurance level 3 and 4, which requires strong mechanisms for security and authentication. Government issued MinID is an example of a requirement to satisfy assurance level 4.

6.6.3 Comparison

The Australian framework covers a broader area of authentication compared to the Norwegian. While the Norwegian only suggest assurance levels and its risk levels, the Australian offers a methodology for information risk management and registration policy, and a large list over areas of requirements that has to be met.

The Norwegian framework compared to the Australian is only a guide that provides instructions and recommendations regarding assurance levels. The task of risk analyses and assessment is up to each institution and then place their solutions at the right level.

Although the Norwegian framework does not mention a lot anything about registration, which is an important part of authentication, the data quality of the users is ensured by the national person register in Norway and distribution security. This could mean that the Norwegian framework either wants to put more responsibility on their parties or that they have greater confidence in their parties.

6.7 Data quality assessment

Each dataset contains some errors, and it is not certain that the errors are discovered at all. To analyze the data sets and try to find these errors are not always easy, and sometimes an impossible task. Correction of errors in data and eliminate the errors can be a time-consuming and tedious process, but can not be ignored or postponed.

"'Source systems shall include accurate, consistent and updated information. This is necessary for the services that use the selected identity management solution, to be able to rely on the information they receive."' (Translated from Norwegian)[1]

A data quality assessment can help us to determine inaccurate, incomplete or unreasonable data and checking of orphaned and duplicated data.

The need for data cleaning is centered around improving the quality of information used by the systems, services and applications in an organization. The services that use the information should be able to rely on information derived from authoritative sources that have inspected the information for errors, inconsistencies and that the data follow a certain format.

We should quality check for errors and incomplete data should also clean up the roles and identities in multiple data sources, remove or disable orphaned accounts or system accounts that are not in use anymore. Clean up external user accounts that are no longer in use or no longer have an approved owner should either be disabled or deleted. Data cleaning is not a one-time job, but rather a continuous process. Data should always be checked for errors, be kept updated and removed when they are no longer required or needed.

6.8 Usability

Although usability is a wide field related to almost everything regarding computer systems, but the part we are most interested in this thesis is the

web aspect.

Usability is one of those areas of web development which is very important, obvious when you look at it; and easily overlooked if you're not actively looking for it. It is simply the question of whether the site is easy to use. From an usability perspective, the question becomes: "Are the features which are needed present, easy to find, and easy to understand?"

One of the most basic issues for web usability is whether the interface is similar to interfaces the user has seen before. There are certain standard paradigms of web interface design in how navigation is presented, where it is located, and what it does. There are expectations formed by thousands of web site interactions which lead users to quickly and easily understand similar interactions. Designers frequently want to "break out of the box", which is fine, but only as long as they understand and prepare for the risks.

In addition to maintaining a common structural paradigm, usability requires developers to anticipate the user's needs. To take a basic web site login as an example, good usability requires that the web site:

- Is easy to find.
- Informs the user of the advantages of registration.
- Notifies the user of your use of their private data whether you use it or not.
- Makes it easy for the user to be aware whether they are currently logged in.
- Makes it easy for the user to log out.

These features are not absolutely necessary to have a functional registration and log in system, but they will make the system significantly easier to use and more successful for the user.

Usability also extends to the structure of the page as a whole. Huge, long blocks of text are more difficult to follow and read - breaking them up into sections delineated by headings, lists, and images can make the document as a whole much easier to digest.

Usability is a broad issue for web interface design and security. One of the greatest challenges with usability is its priority in projects. The assessment of a good user interface should play a bigger role under the preliminary design of the system. Considering usability throughout the design of a site can make a huge difference in making sure the end product meets everybody's needs in design, function and security [12].

Usability Principles

Josang et. al. [21] state that: "Usability of security is an extremely important, but still poorly understood". The paper tries to get a more systematic overview of usability, by introducing some security usability principles. They distinguish between a security action and a security conclusion in a scenario when the user is directly using an identity system. **A security action** is something the user actively does, this can for example be to type in a password or agreeing to a policy. **A security conclusion** is something the user passively does, like observing the security indicators given by the system or to recognize an URL.

The usability principles proposed related to security actions and security conclusions are described below.

1. *Security Action Usability Principles*
   (a) *The users must understand which security actions are required of them.*
   (b) *The users must have sufficient knowledge and the practical ability to make the correct security action.*
   (c) *The mental and physical load of a security action must be tolerable.*
   (d) *The mental and physical load of making repeated security actions for any practical number of transactions must be tolerable.*
2. *Security Conclusion Usability Principles*
   (a) *The user must understand the security conclusion that is required for making an informed decision. This means that users must understand what is required of them to support a secure transaction.*
   (b) *The system must provide the user with sufficient information for deriving the security conclusion. This means that it must be logically possible to derive the security conclusion from the information provided.*
   (c) *The mental load of deriving the security conclusion must be tolerable.*
   (d) *The mental load of deriving security conclusions for any practical number of service access instances must be tolerable.*

7 Identity management solutions

7.1 Federated solution

FEIDE is chosen because it is the identity management solution at University of Oslo and because of its extensive repository of documentation. FEIDE's authentication method is regular username and password, which suits the purpose of the following evaluation.

7.1.1 FEIDE

FEIDE (**F**ederated **E**lectronic **IDE**ntity) is the centralized identity management solution for the educational sector in Norway. Every student and employee receives one electronic identity with an username and password that can be used at a number of services. FEIDE allows users to identify themselves via a common login service, and determine the requirements for how personal data should be handled by the individual organization. All the affiliated institutions has to trust each others data, based on that FEIDE trust that each institution continually deliver accurate and current data. The organizations themselves keep track of personal data about their users, and the largest job during the preliminary FEIDE-implementation is to organize personal records and implement procedures to ensure good data quality. Since person information is local and the information provided is related to an authoritative source, the information has to be maintained in only one place. Making it easier to provide correct and updated information.

In the FEIDE federation there is a single IdP - the central login service. This means that all the home organizations share one common IdP, a centralized model. All SPs only connects to this IdP. This makes it very easy for SPs to reach many users by only connecting with a single IdP. See Figure 5.

FEIDE's central authentication is handled by Moria [34]. Moria is an authentication service for Web-based services. The latest version of Moria is Moria III, based on Sun Access Manager. With Moria III FEIDE uses SAML 2.0 for authorization and authentication. Web services can use Moria instead of implementing a login mechanism themselves, and to reduce development cost and provide high security.

When a user wants to use a service, the service redirects the user to Moria for authentication. If the user is already authenticated, Moria will redirect the user back to service, and send information about the user. If the user is not yet authenticated, Moria opens a dialog where the user provides their username and password, and Moria verify this with data from the user host organization before redirecting the user back to the service. So the user

Figure 5: Feide architecture. [35]

data is still maintained by each organizations. The organizations handles the data through their own "User Administrative System" (BAS).

Figure 6 shows us how the flow in FEIDE looks like from an user's perspective.

7.2 User-centric solutions

When it comes to technologies in the user-centric world, we have two interesting solutions. On one hand, there is OpenID, a solution that is free and open. OpenID is decentralized and gives you the opportunity to choose who you want to trust with your credentials, but the downsides in its core implementation is phishing and usability under authentication. The other solution is primarily known by its commercial version, CardSpace from Microsoft, the idea is information cards containing credentials and presented in a identity selector very similar to your wallet. The information cards are like bookmarks, which can detect sites with a different URL than the one under establishment of the card and can be used to avoid a phishing attack. Unlike OpenID, which is web-based and platform independent, information cards requires client-side software and some infrastructural changes.

Despite their differences, OpenID and Information Cards both provide the same top-level benefits to users and websites:

- Simplified login reduces the many confusing username/password options users navigate today to a few secure methods standardized

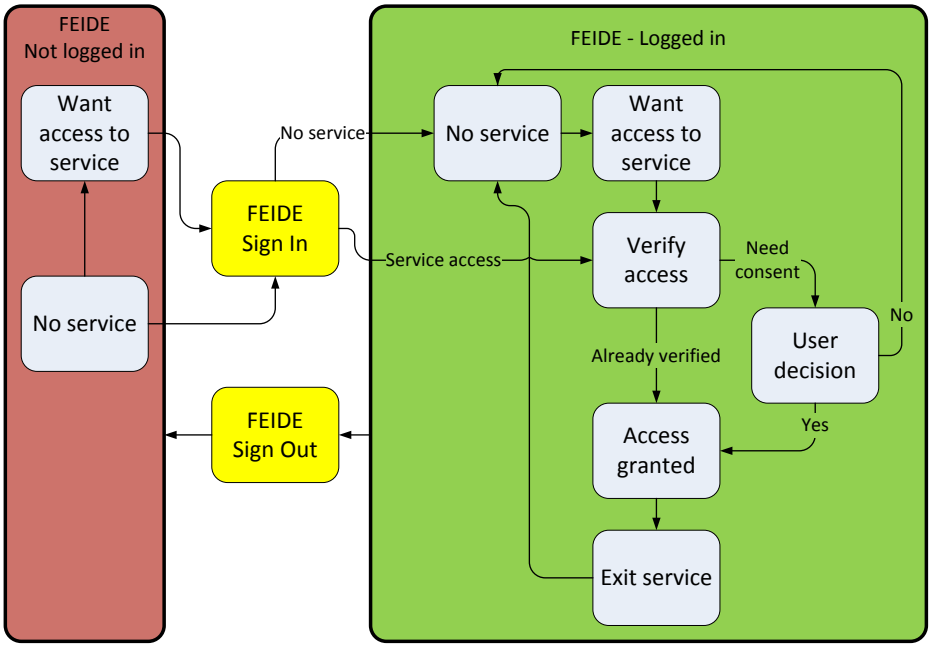

Figure 6: FEIDE flow (user perspective)

across all sites.

- Identity portability lets users ”carry” the same identity credentials across different websites and services, just as people can now keep the same cell phone number across different wireless carriers.
- Automatic data exchange lets users register at a website or fill out a web form as easily as they swipe a credit card to make a payment today.

Based on a profile created by ICAM [15], OpenID is only supported for use on assurance level 1. InfoCard is permitted for Assurance levels 1-3 according to [14].

7.2.1 OpenID

is a decentralized digital identity system that gives all user IDs in URLs (like blogs or homepages). As of now, the current version of OpenID Authentication is at 2.0. From the beginning, OpenID was designed for very simple environments, so it is not complicated compared to other ID services. OpenID is sometimes referred to as a ”Decentralized Single Sign-on for the Web”.

The OpenID model consist of two entities, the OpenID-provider (similar to IdP) and Relying Party (similar to SP). We will hereby refer to a relying party as SP.

Users in OpenID do not need to create and manage a new account in order to sign in to any OpenID-enabled websites (SP). Instead, they need to subscribe to an OpenID Provider, i.e. a website which is trustworthy and provides an OpenID service, and handles authentication. The OP can verify the user's ownership of a relevant ID to the SPs.

Thereafter, users do not need to type in their name, address and birthday yet again and struggle to manage countless IDs and passwords. Thus, companies using OpenID-enabled sites need to make a page for OpenID instead of a page for Sign-up. In addition, they can save costs in managing the users' IDs and passwords, and outsource user authentication services and SSO (Single Sign On) services.

Within OpenID, users are able to assert control over a certain identifier (typically an URL or an XRI) to identify themselves to service providers, called Relying Party.

OpenID's latest version, 2.0, has a very useful privacy feature called directed identity. Directed identity gives the user the ability to create different identifiers for every SP, and that they all point to the same identity (Something only your OpenID-provider would know). This is a great feature in terms of privacy and make user usage aggregation by a third party a little bit harder. Another word for this kind of URL is, opaque. An opaque URL is one that does not itself reveal any information about the user it identifies.

OpenID provider driven identifier selection is another feature. This feature gives the user the ability to enter the URL of their OpenID provider at their SP rather than their long personal and "hard to remember" URL. Google and AOL (American OnLine) has used this feature to let users only type in "google.com" or "aol.com" at their SP. The SP would send the user to their e-mail login screen, which allows the user to login to something familiar. This is an easy way for users to adopt and use OpenID, since most user already have an e-mail account.

Figure 7, which is inspired by a model by Josang [20], shows us how the flow in OpenID looks like from an user's perspective. A flow in Figure 7 could be like: The user want access to a service and the service redirect the user to OpenID-provider for authentication. Since the user already wants to use a service, the service provider asks the user for her consent. If she visits the site for the first time, she has to agree to send some information.

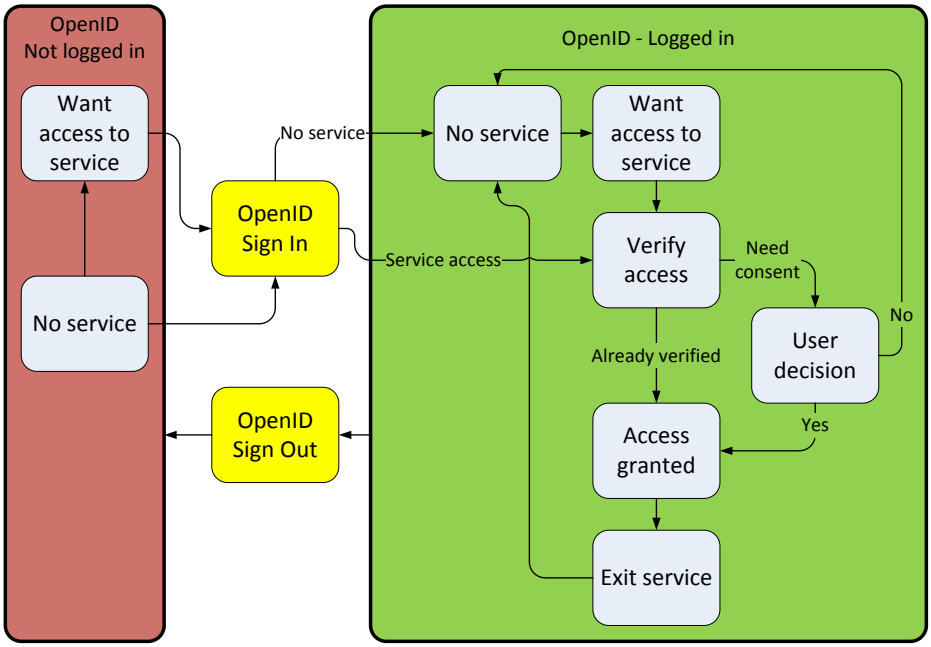

Figure 7: OpenID Flow (user perspective)

If the user has visited the site before, given her consent and agreed to let her provider send the same information every time she wants access, the service is available right after authentication (Her access is already verified). The user can access as many services she wants, as long the OpenID-session is active.

OpenID has been getting a lot of attention from big commercial companies (Yahoo, Microsoft etc.) and some recent interest from some government (for instance US government).

OpenID and phishing OpenID does not enforce any specific form of authentication; therefore an OpenID Identity Provider can use different types of authentication, including stronger ones. This last point in particular addresses one of the main criticism towards OpenID in its early version, namely the low level of security and easy possibility of phishing.

A phisher can create a page that is identical to the users identity provider and redirect her to it. Once the phisher obtains the users credentials, the phisher has access to everything connected that OpenID-provider. This mistake has to happen only once. The source of the problem lies in the http-redirect from service provider to identity provider.

One scenario where phishing is a problem is when the user is being redirected under authentication process, a malicious attacker can redirect the user to a fake OpenID-Provider.

The following use case is central to the issue:

1. User visits a malicious RP page containing what looks like a regular OpenID login form.
2. User enters OpenID URL
3. Malicious RP redirects user to another page that looks like the user's OP (call this Fake-OP)
4. Fake-OP asks user for password
5. User not noticing the difference from his usual OP, enters her password
6. Fake-OP now has user's password.

OpenID in it self can not do anything about phishing attack, the reason is that it does not require any client-side software. As long as you have a browser, you are able to use OpenID. Although there are browser-plugins, bookmarks and other ideas on how to solve the phishing problem, these are only add-on features. The user has to choose their way of protecting themselves.

Although OpenID reduces the number of login screens presented to the user to only one familiar, a normal user would not know the difference between an original and a phishing site. This assumption is based on the user not being properly educated about how to avoid being phished. This honey pot problem is a hot topic in the OpenID community. There are some interesting mitigations on how to make OpenID more resistant to phishing, like the cooperation with CardSpace. Something we will have a look at later.

A lot of work has been put into improving OpenID, like security and usability issues. Like OpenID User Experience Summit and the OpenID foundation[6]. Their aim is to enabling, promoting and protecting OpenID technologies.

7.2.2 Information Cards

Information Cards are personal digital identity cards that people can use online. Visually, each Information Card has a card-shaped picture and a card name associated with it that enables people to organize their digital

[6]http://openid.net/foundation/

identities and to easily select one they want to use for any given interaction. The Information Card metaphor is implemented by identity selectors like Windows CardSpace, DigitalMe or Higgins Identity Selector. We will focus on the Windows CardSpace in this thesis, to illustrate some of the functions available with Information Cards.

There are two main elements in Information Card model; IdP and SP (often referred to as relying party, but we will use the term SP to avoid confusion). The IdP works like a Security Token Service, which is a web service which issues and validates security tokens.

Using Information Cards, users can authenticate without the need for an username and password for every web site. At sites that accept Information Cards, users can choose a card from their identity selector (see figure 8) and get authenticated. Keyloggers are not able to snoop any password from the user. A card can be used at multiple sites.

Each Information Card utilizes a distinct pair-wise digital key for every realm where a key is requested. A realm may be a single site or a set of related sites all sharing the same target scope information when requesting an Information Card. The use of distinct pair-wise keys per realm means that even if a person is tricked into logging into an imposer site with an Information Card, a different key would be used at that site than the site that the imposer was trying to impersonate; no shared secret is released.

Furthermore, many Identity Selectors provide another phishing detection, where the HTTPS certificate of the Relying Party site is checked and compared against a list of the sites at which the user has previously used an Information Card. When a new site is visited, the user is informed that they have not previously used a card there.

SourceID were working on an plugin for Apache for Information Cards, but it seems to lack any further progress for the last three years [19]. A very exciting idea that could expand the adaption of Information Cards.

There is also a foundation, called the Information Card Foundation[7], with members from Deustche Telecom, Equifax, Google, Intel, Microsoft, Novell, Oracle, and PayPal. Their aim is to promote the adoption of Information cards, and to build a better digital identity system for the Internet that enables people to easily and safely share identity information across all web sites and services.

[7] http://informationcard.net/foundation

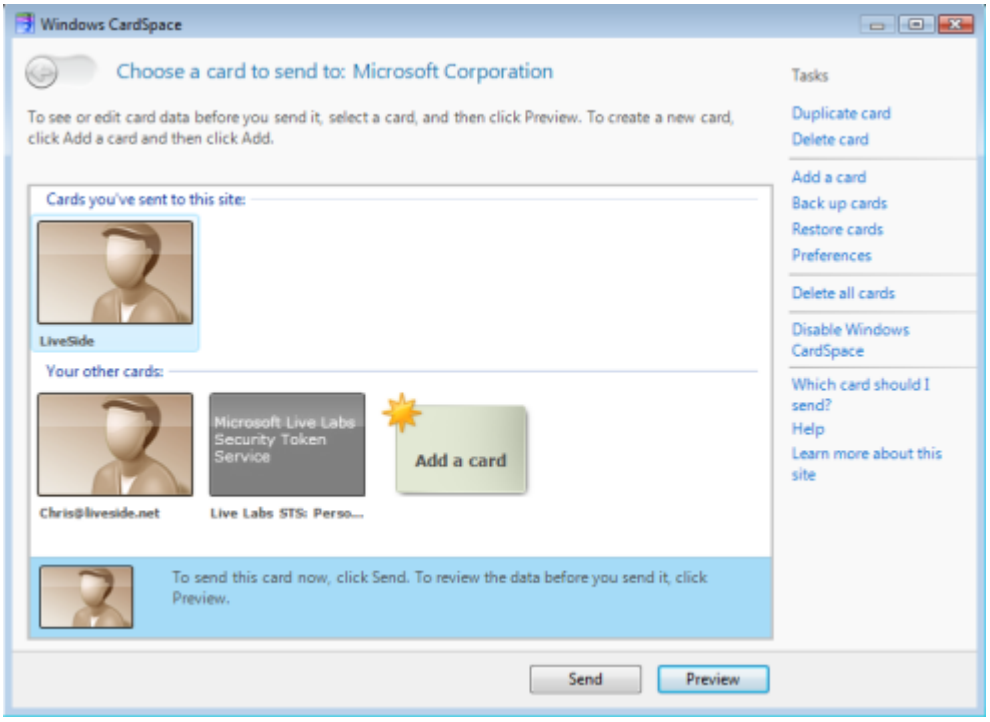

Figure 8: CardSpace Identity Selector

In principle, Information Cards could be easily integrated into federations as an additional authentication method.

CardSpace Windows CardSpace is a product of Microsoft and their solution of supporting Information Cards. CardSpace is a client application that comes pre-installed with the Windows .NET 3.x Framework. The CardSpace runs on the desktop as an identity selector which holds Information Cards. There are two types of Information Cards supported by CardSpace: Managed cards and Personal cards (also called self-issued cards).

Managed cards are cards that an Identity Provider has given to the user, who has imported it into Identity Selector. These cards contains metadata with reference to the IdP, the actual information is at the IdP. So in order for them to be used, the selector would have to send the policy request to the IdP which will send the response back, so that the user can review the information and give her consent.

Self-issued cards are cards where the user is acting as the IdP and provides all the values for the claims. CardSpace provides the facility for the user to create, edit, export, and import Personal Cards. The data for these

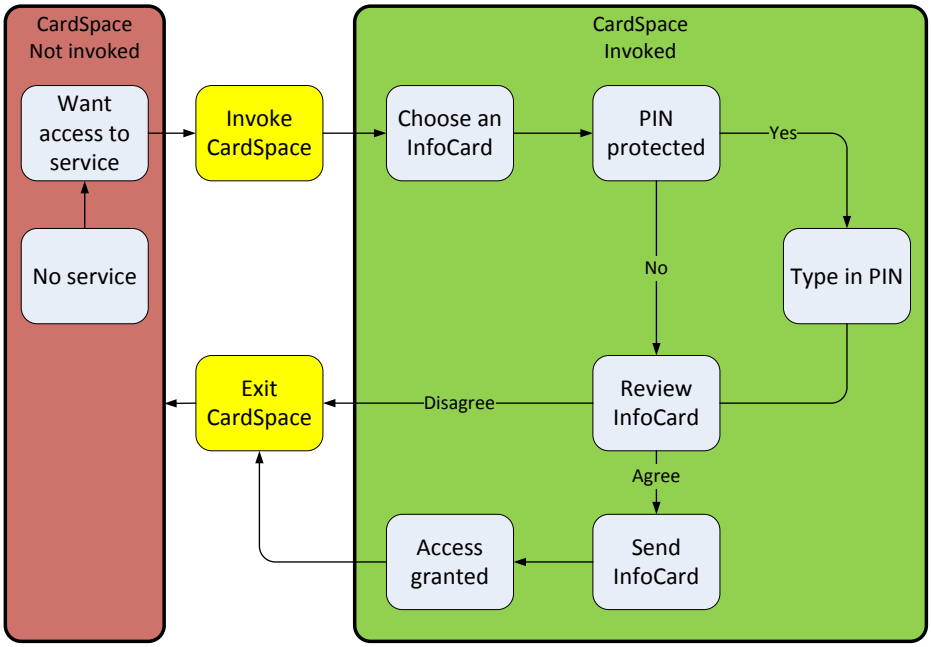

Figure 9: CardSpace Flow (user perspective)

cards is encrypted and stored on the user's computer. The claims that a Personal Card can support are fixed, so that SP can accept a common, consistent Information Card.

In a typical CardSpace scenario (see figure 9), users wants to access a site that provides a service with their browser. The SP's authentication page contains an icon to indicate that it supports Information Cards. Upon clicking this icon, the SP sends an authentication challenge to the user's browser. Included in this challenge is a determination of which attributes of a user's identity are required (these required attributes are sometimes called claims). The identity selector is displayed on the user's desktop, asking the user to choose which information card to use to authenticate to the SP. The set of required and optional claims that are requested by the SP is displayed so that the user is aware of what information is being requested. Additionally, the user has control over which information card to use and whether optional attributes are provided. There can be a PIN in order to use the card, which the user can set herself. After an information card is chosen The browser then sends a security token to the SP where it is validated to complete the authentication flow.

Issues with CardSpace: As long as the user uses managed cards issued by an IdP, there is a chance that an evil IdP could sell your usage information to a third party. Especially information about the SPs you are visiting.

8 Method for evaluation

We want to evaluate privacy, usability and trust from users perspective. We will also look at some security protocols visible to the user during login and log-out.

The solutions will be analyzed based on what the user can observe while using the system. The system will maybe state that it supports SAML 2.0, SSO or ensures the safety the data. But this is not always something an user is capable to observe by simply looking at a web site. What we want to do is to determine the security, trust, privacy and usability based on what the solution is telling or showing the user before registration and during usage. This is an attempt to map the user experience from a systematic point of view. An example of this is a scenario where a hotel is promising for example a nice bed, good security, swimming pool and panoramic view. But the moment you enter the building, you realize that nothing was the way you imagined. The metaphor we are after here is that we want to analyze the identity solutions as if we were the hotel guest. We want to look at the ceiling, the quality of the bed and how everything works through our own eyes. Basically, we want to look at the amount of information provided, the behavior of the authentication process and if the solution is easy to adapt.

We will evaluate the solutions based on these scenarios from an user's perspective:

8.1 Scenarios to be evaluated

- Enrollment
- Login
- Logout
- Deletion

We will try to map the most basic features provided by the solutions by analyzing these scenarios. The analysis of logins and logouts will more in depth, since these scenarios are more frequently used. Changes in these scenarios are more visible to the users and affects them more, compared to enrollment (which is probably done once) and deletion (also done once).

Our findings and observation will be categorized into four aspects; security, privacy, usability and trust.

8.2 Security

Federated identity can offer better service at a lower cost, but it also entails some security risks. First, federated identity involves crossing security domains. Ideally, all parties should secure their communication channels against replay attacks, man-in-the-middle attacks, session hijacking, and other threats that allow malicious use of user information. In an HTTP context, security architects consider Secure Sockets Layer/Transport Layer Security (SSL/TLS) with mutual authentication as a security baseline. Still, application developers often avoid, overlook, or only partially implement this step.

User authentication is another weak link in the Web identity chain. Currently, most sites rely on username/password pairs because this method poses the smallest initial burden for users and site administrators.

For SPs, federated identity is less expensive than implementing a high-quality authentication infrastructure because it off loads the authentication task to an IdP. However, SSO could potentially magnify the costs of a stolen password because it expands the scope of malicious activity. Most SSO protocols offer ways to mitigate this risk. For example, they might limit to a minute or less the valid lifetime of the security token that an IdP sends to SPs; some protocols also offer a Single Log-Out (SLO) feature that offers users to simultaneous sign-out of all SSO-accessed Web sites.

The agenda behind this section is to observe the visible behavior of the security protocols and their consistency in the interaction with users.

8.3 Privacy

"Sharing personally identifiable information is a great concern in managing privacy, protecting data, and complying with regulations. With federated identity, however, sharing such information is often a key goal, which raises interesting privacy issues." [23]

In a proper implemented identity management system, the exposure of personal information has to be limited to a minimum [21]. The agenda behind minimizing the exposure is to ensure a user's privacy and only give sufficient information in a transaction so that a relying party can make a decision. It is important to let users be a part of their own identity information exposure. User involvement does not only make users more aware of what is happening, but it could also make them feel more responsible and thus more willing to make the right choices. It is also important to be aware of the amount of information the solution stores about an user. Although

exposure has to be minimized, it is equally important to know how much information the IdP saves about your usage.

The privacy of an user has to be protected throughout the identity life cycle. The disposal of the identity is equally important as the enrollment process. Ability to delete an identity is very important when e.g. the user disagrees to a policy or no longer wants to be a part of the community/service provided.

Anonymity and pseudonymity

A user may want to appear anonymous at a service provider when posting a comment on a blog. This can be solved by offering pseudonymous identifier to the user. The identity provider creates a unique identifier that has nothing to do with a user's real attributes, and connects these to different service providers. The user would identify herself as *123eheiwq* to one service and *456ewqeqw* to another service, and the service provider would know that these identifiers were legitimate. To achieve this function, the framework has implement this as a service to the users [27].

The agenda here is to see how the solutions handle user consent and how much information that is stored about the user and their usage. The solution will also be examined to see if they offer anonymity and how to use it.

8.4 Usability

Based on the usability principles presented by Josang [21], we want to see some of the security actions and security conclusions needed in order to make the right decisions. The aspects we are interested in are user adoption, usage and user-knowledge.

Adoption:

We want to know the solutions pre-requirements in terms of adoption. What does the user have to do before they use the system? Do they have to install some software? Is the experience similar to something they previous know? Is this a new concept? Portability is also analyzed.

Usage:

We want to have a look at the mental and physical load required to login and logout of the system. We will also analyze the feedback and information

provided to the user.

User-knowledge:

How well does the user need to know the system in order to use it securely? Does the feedback alone provide sufficient information or does the user have to do some research first. Although the SSO experience is transparent, there can be some behavior that confuses the user. We want to map these irregularities and see if something can be done to mitigate the issues.

8.5 Trust

The goal of the metric regarding trust is not to look at the business intentions behind federations, since this is outside the scope of this thesis. We will have a look at some of the aspects that could have an effect on users trust when using the solution.

Trust on internet is usually something the user will have to determine based on the security indicators and behavior given by the web site. In many cases the only possibility for the user to determine a site's credibility is to look at the visual information, policies or something familiar. The user will also have to look at the security behind the solution, but this is not always feasible.

Consistent feedback:

Users need sufficient information in order to make the right decisions. If users feel that the solution is not helping them to do that, it could have a negative effect their trust. That is why consistent feedback is important to let the users know that the system care about their safety and privacy.

Easy design:

The solution should have an interface with easy and intuitive navigation. There should be easy access to information or an easy way to contact the administrators. Professional design is a factor that has an effect on users trust. What we mean by a professional design, is that the different dialog screen (like login) provide the needed information and nothing else (e.g. ads for something entirely else).

9 Evaluation of solutions

The main focus of this evaluation is to see two different ways the user can authenticate themselves and have a look at some of the potential problems in the user experience. The problems will be analyzed and some suggestions will be made if possible. We will pay attention to security, privacy, trust and usability based on visible information available during usage and use theoretical information where it is needed. Basically, what the user can see and have to understand before, under and after the identity's life cycle. The section is summed up by a discussion on possible mitigations based on the findings.

We have chosen to evaluate a federated solution with regular username and password, and an user-centric solution to see if CardSpace can bring something different in terms of authentication.

On one hand, a standard federated scenario with username and password will be shown using FEIDE. This is to explore the basic features like consent, consent history and choosing home affiliation (IdP discovery). On the other hand, we will have a look at a scenario where we combine CardSpace with OpenID to access OpenID-enabled services. Microsoft are working on an OpenID-feature [7] within CardSpace, but this feature is not available yet. The collaboration with Microsoft and OpenID is to improve OpenID's usability and security, and expand usage of CardSpace. The reason we will explore the combination of OpenID and CardSpace is because they complement each other in terms of security and convenience. CardSpace has the benefit of presenting credentials in a user-friendly way, a model very similar to peoples wallet today. CardSpace has also been proposed as a mitigation against some of the phishing problems with OpenID. While OpenID's has the benefit of being broadly accepted and implemented because of its simplicity compared to CardSpace.

Although the observation is intended to be as objective as possible, some subjectivity is unavoidable in an evaluation like this. The evaluation is based on documentation and real-world usage of the solutions.

9.1 FEIDE

FEIDE acts like a mediator (proxy) for users that wants to communicate with services and home organizations to vouch for them. Authentication is handled by the IdP (home organizations) and authorization is left to the individual service providers. FEIDE gives the user a consistent user interface and passes the user input like username, password and consent to the right parties. The personal information (when the user has given her consent) is

sent through FEIDE as an assertion from the home organization to the SP. The transaction is done using SAML 2.0.

FEIDE will have less scenarios to evaluate since the users do not create their own identity themselves, the information is created in cooperation with the Norwegian government person register.

9.1.1 Scenarios

Enrollment The users in FEIDE are either students or employees, and they are locally enrolled at their organizations. Students are automatically enrolled to the system when they are accepted to a university or a college. Employees are also enrolled as soon they start at their jobs. Once the identity is created at the users home organization, the user is given a user name and password and given right permissions. FEIDE does not store any personal information, the information is stored at the users home organization (e.g. University of Oslo).

Since the user does not see anything from the enrollment process, we will not be able to evaluate the process here. The users assume that the information about them is correct (like username, full name etc.). It is up to the user to report to their home organization's help desk if something is incorrect. The registration process[8] at the universities/colleges is rather extensive, based on the amount of information needed to be enrolled as for example, a student. A person has to at least be a Norwegian citizen or have a residence permit to apply. This means that the registration is backed up by some government papers/information. When she has been accepted to the her university/college and paid her intuition, the identity is created. So the claims made by the home organizations in FEIDE are very strong, based on the extensive registration process.

Login When a user logs in to a service for the first time with his FEIDE identity, the following steps take place:

1. The user wants to access a service that requires login.
2. The service asks FEIDE whether the user is already logged in. If the user is not logged in, FEIDE's login window appears. If the user is already logged on to a FEIDE service, the user does not need to enter her username and password due to Single Sign On (SSO).
3. The user enters her username and password. She also have to check that the right affiliation is chosen.

[8] http://www.samordnaopptak.no/info/english/

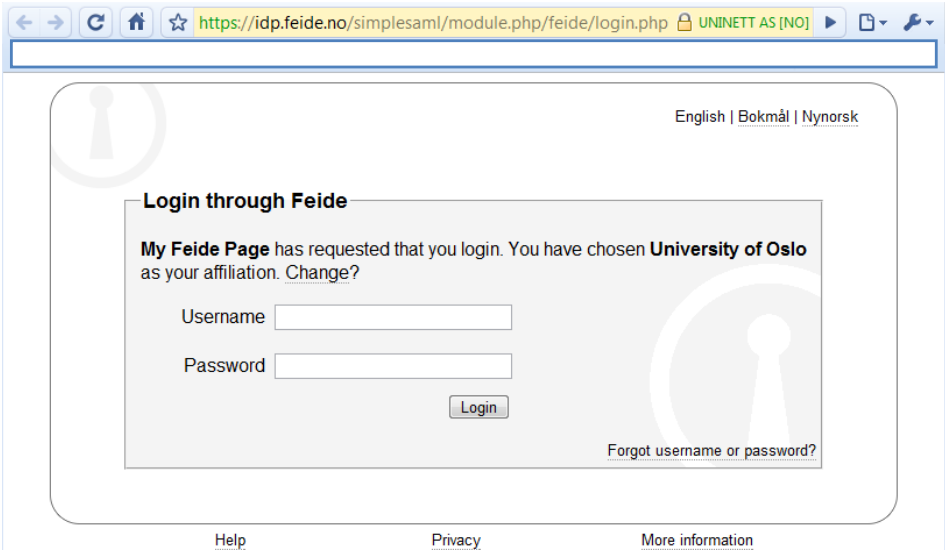

Figure 10: FEIDE login screen

4. FEIDE send this on to the user's host organization for control.

5. If the user name and password is correct, the host organization sends a confirmation to FEIDE. In addition, the personal information requested is sent from the host organization. All control of usernames and passwords are handled at users host organization. FEIDE does not store any information about users, but takes the responsibility of a message mediator between the host organization and services.

6. The user must verify that she agrees to the personal information that is going to be forwarded to the service. This is only done once if the user agrees to approve this action in the future. If the policy changes, the user will have to renegotiate the consent.

7. After the consent is given, FEIDE notify the service that the user is authenticated and sends data to the service. The service will only receive the personal information it has agreed to receive from FEIDE.

8. The user will then get into the service. FEIDE only handles the authentication, it is up to the service to give the user access based on the information given by the user.

The feedback from the system is consistent and the user is involved in the transaction process, although a lot is happening behind the scenes.

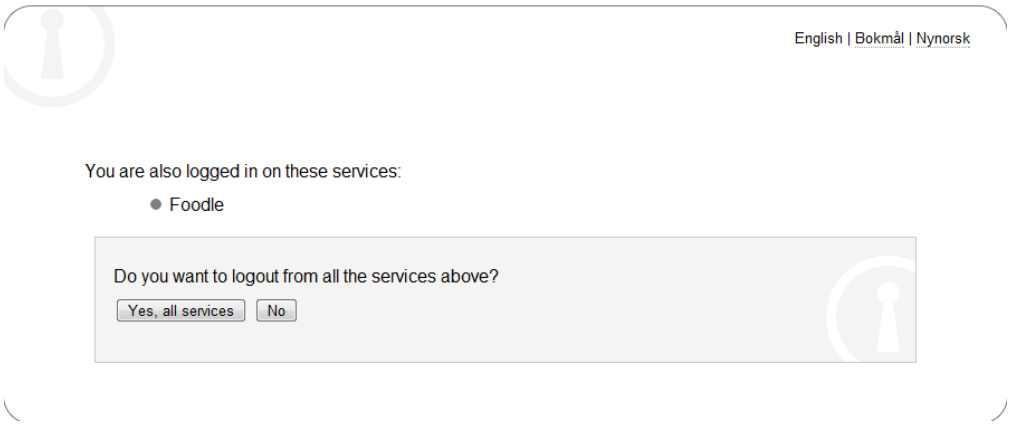

Figure 11: Feide SLO

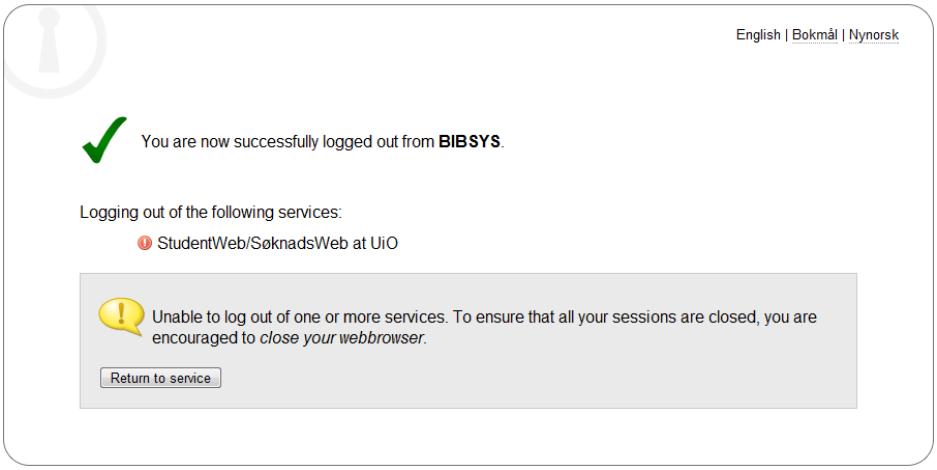

Figure 12: Feide SLO - Partially logged out

Logout A logout is done with the help of SLO (Single logout), which means that you are logged out of all services connected to the same federation. See figure 11 and 12 for examples of FEIDE SLO screens.

1. The user logs out of a FEIDE service by clicking on the "Sign out" button.
2. A dialog appears asking whether she wants to log out of the other FEIDE services as well. (See Figure 11)
3. The user can then select to log out from one service or all of them. If the user only chooses to log out from one service, the user will still be logged on the other services, but the user will however have to log in

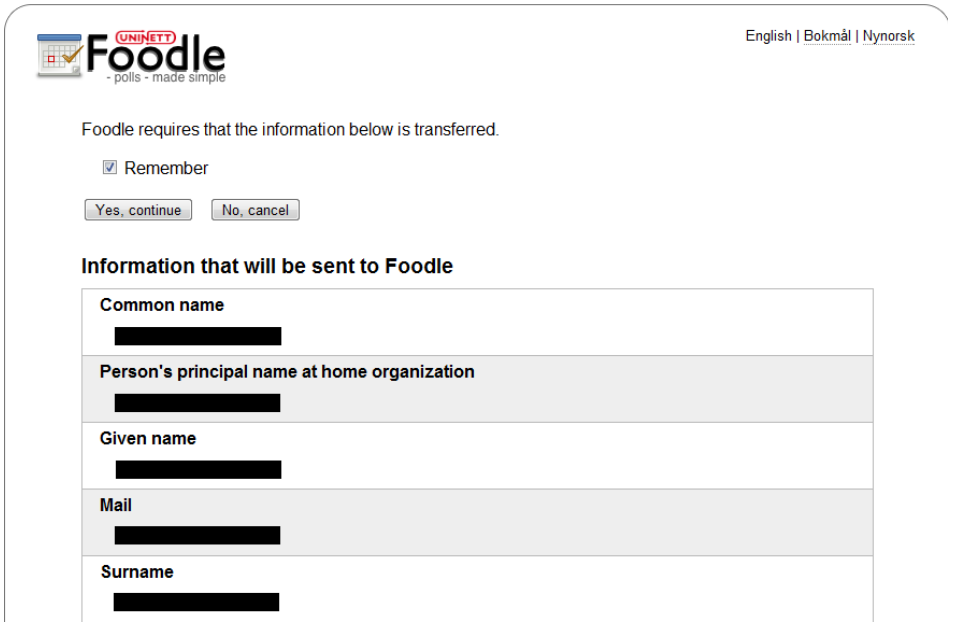

Figure 13: FEIDE consent under login

again if she wishes to enter into a new service. The ability of Single Sign On disappears when she decides to logout, even though she decides to log out from one service.

Maintainance/Use We have already been introduced to the consent function provided in FEIDE. The user gets a list over the information requested from the service provider during login to that service, and the user can choose to give her consent or not. The user can choose to give consent once or always, this agreement can be canceled at any time through FEIDE's "My Page". The information that are sent is not stored at the service, this is only information needed to verify the users. But this is not something the user can conclude by just looking at the consent screen. The SAML 2.0 protocol is very strict about the policy and the parties involved has to agree on what attributes are released from one party to another party. Based on this, FEIDE ensures that only the needed information is exchanged and nothing else.

Even though a federation is about gathering all your identities and using just one unique, an observation tells us that the users does not know what information the different service providers has about them. Unless the user visits the sites themselves. Some service providers can have a login separated from FEIDE, which is normal. This means that there are two ways to get

access to the same service, which also tells us that the password fatigue problem is only patched. It is not fully mitigated.

Deletion The user account is deleted when the student or employee quits, but the username is never used again. This is handled by the home organization. There are no documentation on whether FEIDE handles provisioning on behalf of the home organizations. But based on the responsibilities given to the parties, deactivation of user accounts is probably something that the home organizations has to handle themselves. We asked FEIDE about deactivation of accounts and were told that this is a process handled by the home organization.

9.1.2 Evaluation

FEIDE requires no client-side application and is only dependent on what the user **IS** and something she **KNOWS**. Since the process of enrollment and deletion is not visible for the users, the high-level descriptions are from the identity process at University of Oslo.We found some issues the user can stumble upon during their usage. Some are noticeable at first sight and some others are note. The issues and observations are sorted and presented in the four following sections.

Security

The architecture of FEIDE is a centralized architecture based on a WAYF (Where Are You From)-model. FEIDE sends the username and password to the user-selected home organization for authentication and does not store any personal information. All the information available about an user is pulled from the home organization at the user's request when needed.

During login to some of the services provided through FEIDE, we noticed some inconsistencies in the SSO experience. FEIDE's "My page" has a list over services offered in through FEIDE. We tried to log on to some of the services utilizing the SSO-functionality, which went fine, but when we tried to log on to StudentWeb (a place to access your student data and course registration) we had to log in again. The home affiliation were still the same, there were also no information on why we had to do it. This could be confusing for the user and since the service is available outside the university, an user could be a victim of a phishing attack. After a quick e-mail to FEIDE, we were told that the reason was due to not fully implemented SSO functionality. But the SSO functionality reduces the number of times the user will have to type in their username and password, which means less exposure and more secure interaction.

FEIDE does not inform the users about the SSO-functionality during login. This could be a problem if FEIDE did not implement SLO as well.

While we were evaluating the logout scenario, some issues appeared. SLO is a great feature, that enhances user experience and makes it harder for malicious users to hijack the authenticated session. But an issue with SLO lies in the browser. When the user decides to log out using SLO, she gets a list over services where she is still logged on and asks if she wants to log out of them as well (See Figure 11). The problem occurs when the user still has services open on other windows/tabs. Although the user is logged out, there is a possibility that a page with some sensitive information is visible. Another issue is when FEIDE is unable to log the user out of a service, a message appears saying that they encourage the user to close the browser (See Figure 12). But when we closed the browser and restarted it, the service was still active[9]. This is because of the browsers ability to store sessions called "session restore", so that the user can continue with her work next time she returns to her desk. This issues is only available for a period of time, because the session will eventually time out. If the user clear out her history before closing the browser, which can be automatically turned on in e.g. Mozilla Firefox, the problem goes away. With that being said, this potential problem is a browser-issue and goes beyond this thesis. The last issue is that FEIDE seemingly prioritize the services in some way. Some services are instantly logged out, but there are few services that require the users to handle this themselves. This could either be to go to that service and click on log out or to close the browser, which could cause some problems as mentioned earlier. The user experience is not consistent. The tested services were BIBSYS[10], StudentWEB[11] and Foodle[12], whereas BIBSYS and StudentWeb are higher "prioritized" than Foodle. After a mail correspondence with FEIDE, we found out that BIBSYS and StudentWeb probably has problems responding to the logout-request. This is mainly due to their lack of support for SLO. FEIDE has tried to implement SLO according to SAML's specification, and addition to that added some feature for better user experience.

SLO does minimize the possibility of being hijacked, but it requires proper implementation and some help from the browsers to utilize its potential. The browsers could clean out the windows/tabs belonging to the same session. But this could raise some new security issues regarding browsers, like pop-ups that closes all your windows to show their ad.

[9]The test was on Mozilla Firefox 3.6.3 and Google Chrome 4.1.249.1045
[10]http://www.bibsys.no/english/pages/index.php
[11]https://studweb.uio.no/as/WebObjects/studentweb2?inst=UiO
[12]https://foodle.feide.no/

Trust

Consistent feedback: From users point of view, the interaction with FEIDE (the IdP) is a very consistent experience. The SP's logo and possibly the affiliation are the two only changes in the interface. Users know who the request is from and this gives the assurance needed to trust FEIDE.

FEIDE's SLO feature is great in terms of building trust. The feature basically says that you do not have to log out of all your active services, they can do it for you. But FEIDE should also tell the user why they are doing it (to minimize the risk of hijacking) and what users should specifically do if some of the log-out fails. The feedback informs the user about the name of the services, but it does not include a link to the service so that the user can log-out themselves. At this point, the most convenient choice is to just close the browser.

Easy design: FEIDE's user interface is simplistic and easy to use. The focus is on the right information at the right time is something they have done very well.

Privacy

There is no history of visits on "My page" other than history of consent. But these consents can be revoked, which means that the user will not have any history of ever visiting the site. This could be an useful feature to include for security purposes (to detect e.g. misuse). It is highly unlikely that FEIDE would sell this information to a third party for profit and lose their reputation. Since FEIDE is a government-approved authentication service.

Anonymity and pseudonymity: FEIDE offers no feature regarding anonymity.

Usability

Adoption: Users only need a web browser to take advantage of FEIDE. The experience is pretty much like a normal username and password scenario, which most users are familiar with.

User-knowledge: Most users would be met with the right affiliation during login, because of the "where are you from" feature, and would not need to use FEIDE beyond authentication. Users are able to get help or education at their schools/university/college.

Usage: Although the SSO experience is transparent for users, FEIDE's SLO solution gives a nice feedback if users decides to sign out properly. The feedback from the SLO solution (See figure 11) is a very good feature, especially in terms of usability.

9.2 OpenID and CardSpace

This combination is an attempt to offer better usability and security for users. The identity selector can remember the your identities and be a tool against phishing (Big problem with OpenID). OpenIDs are shown as visual cards, which is a big improvement for users dealing with several OpenIDs. The bookmark-feature in CardSpace can be utilized to detect malicious pages under login. Microsoft are currently working on a OpenID-support for CardSpace, but it is not release to the public yet. Their solution does not require any implementation from either the OpenID-provider (OP) or service providers (SP). Its function is to associate different OPs to different SPs and remember some information like: where the card was last used, when it was last used or who the OP is.

There are many OpenID-providers to choose from, but in this case we need a provider that supports information cards. Luckily, one of the providers listed on OpenID's homepage[13] offered this support. myOpenID[14] was one of the first OpenID-providers on the market and they have support for multi-factor authentication, which include SSL-certificates, information cards and some other methods.

To make this work, we will have to create an OpenID and then associate a self-issued card to the account with the help of CardSpace.

9.2.1 Scenarios

We are going to evaluate security, trust, privacy and usability based on four stages of the identity life cycle.

Enrollment

myOpenID: Register an OpenID is like register to any other services. You have to chose an username and password, provide an e-mail and pass a captcha-test[15], to prove you are human. Once you confirm your account via e-mail the account is ready to be used. myOpenID guides the user through

[13] www.openid.org
[14] https://www.myopenid.com/
[15] A CAPTCHA is a program that protects websites against bots

Figure 14: myOpenID with Information Card option

a tutorial, which is useful, but the user most likely have a certain service in mind already.

CardSpace: The easiest way to create a self-issued card and then associate it with myOpenID, is to log into myOpenID with a CardSpace enabled browser (The option to sign in with a card will appear. See Figure 14) and add a card to the account. You can manage your information cards under authentication settings. Once CardSpace is revoked, the rest of the screen goes out of focus so that all the attention is on the identity selector (See Figure 15).

Steps inside CardSpace:

1. Choose ”Add a card”
2. We have the choice between Personal and Managed Card. We will make a Personal Card (Self-issued) here.
3. To make the card, Card Name has to be filled. An optional image can be set to make the card more familiar. Some optional data, like first name, last name and e-mail, can be included in the card. The last action is to click on ”Send”. Click on ”Send”.
4. In the next step, CardSpace lets the user review the card before it is sent. You will where the card is going to be sent (URL), a Site-specific card ID is shown, whether the card has been use at that particular

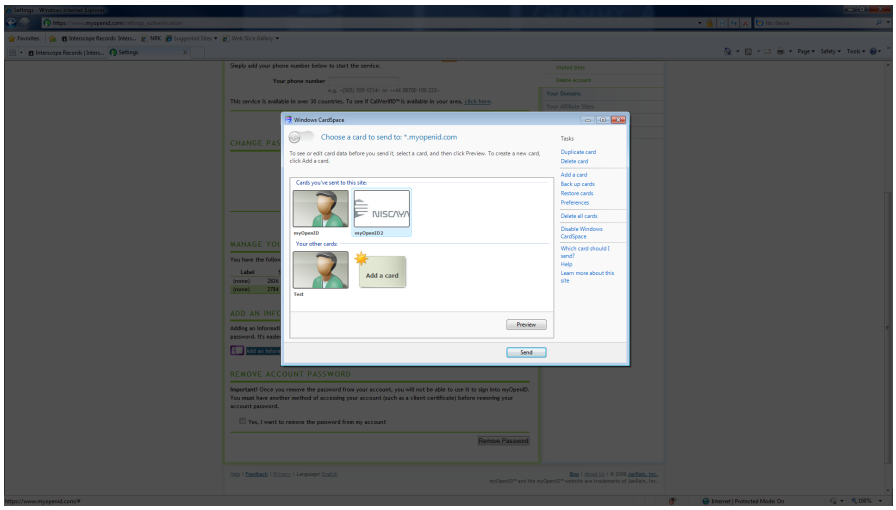

Figure 15: CardSpace in focus

site and when the card was created. Optional data like first name, last name and e-mail can be included in the transaction. The user can also have a look at the card usage history and set a PIN[16]-code on the card. The last action is to click on "Send".

5. The card is then accepted by myOpenID

The card shows up in myOpenID with the information about when the card was first used and the option to revoke the card. At this point, myOpenID lets the user remove their account password (See Figure 16). This means that the information card you have associated with myOpenID, is the only way get access to you OpenID. The possibility of resetting your password via e-mail is gone.

Register/Login Since the OpenID is not integrated into CardSpace, we still have to manually type in our OpenID-providers URL (myopenid.com) at service provider. But nevertheless, using CardSpace with OpenID is a step towards improving the user experience. Hopefully, we will be sent to the right OpenID-provider, where we can choose to login using an information card. If we were sent to a malicious site, the card would state that it had never been used at that page before. This information should make the user a little bit suspicious and the URL would be something different. A nice feature would be to just choose your OpenID card and the CardSpace would handle the authentication for you. The behavior of this idea is similar to the experience with password managers in browsers. Although the behavior is similar, the password manager would type in your username and password

[16]Personal Identification Number (PIN)

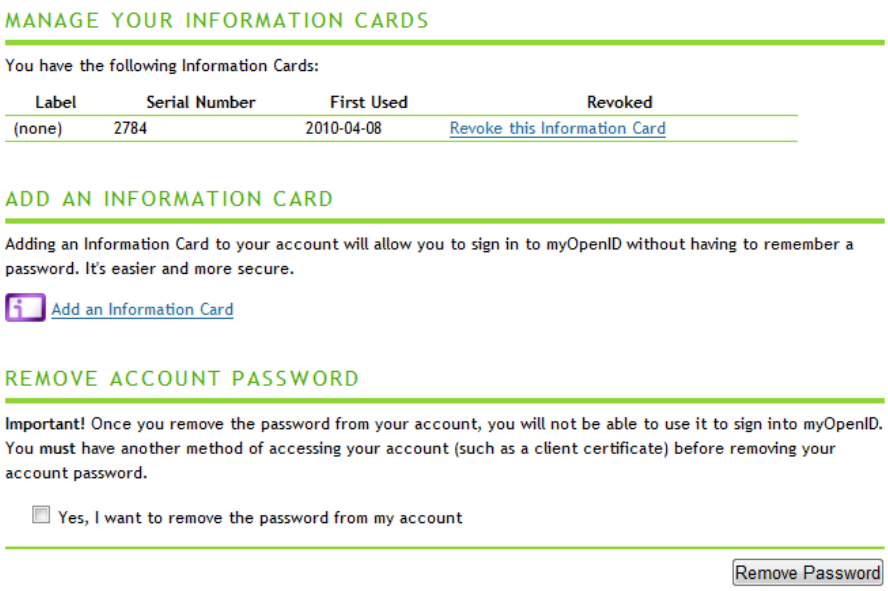

Figure 16: MyOpenID - Removing account password

in the form, while CardSpace would send a cryptographic version of your password. CardSpace could even utilize OpenID's directed identity and not give away the user's real OpenID-identifier.

We have chosen Ziki[17] as our service, to illustrate how we can register to a service and login to our OpenID-provider. Ziki is an online service to help companies find the best service providers to carry out their projects, but this is not important. We will only focus on the registration part and login.

We are not logged in at myOpenID. Starting at Ziki:

1. We want to signup.
2. Ziki offers us to signup using OpenID. We type in "myopenid.com" in the URL box.
3. We are redirected to myopenid.com and chooses to sign in with an information card. Our whole screen focuses on CardSpace (See Figure 15). We chooses our myopenid-associated card and sends it to myopenid.
4. We are asked to give our consent and if we wishes to include one of our registration personas (See Figure 17).

[17] http://www.ziki.com/

Figure 17: myOpenID - Consent and registration persona

5. myOpenID sends us back to Ziki with prefilled information based on our registration persona. We could fill in this manually, but OpenID's attribute exchange has made this process very easy to use. We agree to the terms of use and click on "Create Your Account".

6. On the next screen, Ziki wants us to activate our account via e-mail.

7. We log on to our e-mail account and click on the activation link.

8. The account is created and activated.

Ziki requires e-mail activation, but many other sites do not (like e.g. ClearBits[18] and Koornk[19]). This depends on the services registration process and the reason that Ziki uses e-mail activation could be that they use the same process for OpenID-users and regular users with only e-mail. E-mail activation in addition to OpenID is somewhat overkill (and unnecessary). The reason is that OpenID serves the same purpose in a registration process, which is to prevent bots and that a human is behind and that a valid e-mail address is used.

Logout To log out of an OpenID-provider is not so different than logging out of a service. Click on "Sign out" and you are logged out.

[18] www.clear-bits.net
[19] www.koornk.com

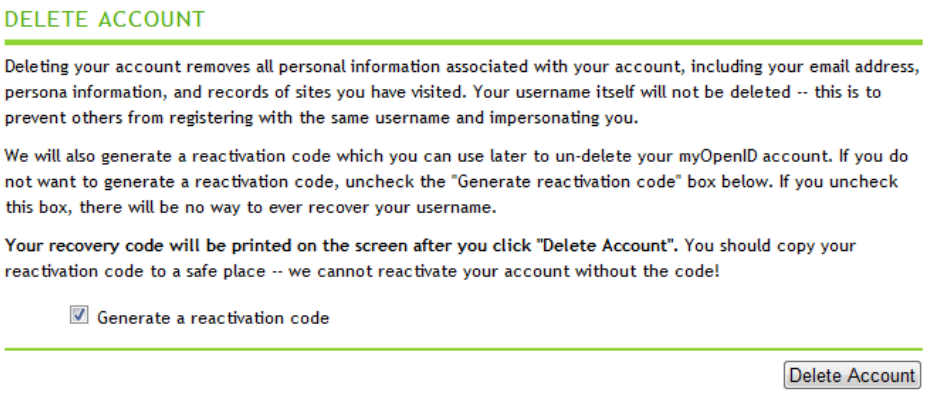

Figure 18: myOpenID - Delete account

OpenID does not support Single Log-Out. The user has explicit click on the log-out button on every service she is on including the OP.

Deletion Figure 18 shows that myOpenID offer users an easy way to delete their account.

To give the user a second chance myOpenID offers to generates a reactivation code, so that the user can reactivate her account. But this is only if the user decides to let myOpenID do so by checking a check-box. The account is impossible to recover without the code.

myOpenID promise that all personal information like history of usage and e-mail address associated with an user will be removed, when the user wants to delete her account. The only information that is not deleted is your username, this is to prevent others to register with the same username and impersonate someone. An important problem associated with hosting your own IdP is for example when the domain expires and someone else buys it, the new owner could immediately login to all her accounts. This is the way OpenID is designed to work, whoever controls the domain is able to authenticate as that URL. But in our case, the user will have to trust that myOpenID will always be there and that no one takes over her URL.

9.2.2 Evaluation

In our scenario, CardSpace works like a password manager and a bookmark checker. It will only log you on to your OpenID-provider and give you some information about the provider you are going to send your card to. Using

CardSpace with OpenID makes the authentication process much easier and in some ways safer. Assuming that the computer is secured.

Although CardSpace can serve as a mitigation against some of OpenID's phishing problems and enhance the user experience. The privacy issue regarding OpenID-providers aggregation of SPs is still there. This is because CardSpace is only implemented on the providers side and not at service providers as well. But if InfoCards were to be implemented at both ends, we would not need the combination. We would probably either deal with InfoCards or OpenID or both in parallel.

This combination does not require any implementational changes for service providers. As long as OpenID-providers support InfoCards, the users can log on without the need of remembering either usernames or passwords. They will only have to remember their provider's URL or bookmark the URL to make it even easier and safer.

We found some issues the user can stumble upon during their usage. Some are noticeable at first sight and some others are note. The issues and observations are sorted and presented in the four following sections.

Security

The first and not always an obvious requirement before an user decides to use a service, is to have a secure computer. A proper implemented identity management system will not do any good if the user is using a compromised system with keyloggers or trojan horses. But this problem is not a discussion we will have here.

Logout and SLO: The OpenID specification does not say anything about SLO. This could be a security risk for users that closes their browsers to log out. Users will have to either log out before closing their windows/tab or have them open until they log out of their OpenID-provider and then click on log out for every service they use. One of the consequences of not having SLO with SSO, is the possibility of hijacking an active session. Imagine that an user uses a browser that stores her tabs (so that she can continue her work next time) or has multiple windows opened. The user decides to only log out of the OP, closes the browser or only the current window and leaves her computer unlocked. Someone else could then reopen her browser or restore her minimized windows and get access to her still active services. The damage can be even larger if the user only logs out of the service and not the OP. The next person on the computer could change the password or e-mail associated with the account, and take over the account. This could

very well happen at someones office, internet cafe or at someones laptop.

The history over visited pages inside myOpenID (See Figure 19) could be used to get an overview over the services you are still logged into by looking at "Last Signin". But this is not something a normal user would do each time she logs out of a session. There are some discussion on OpenID's discussion board about the possibility of Single Log Out (SLO) of some sort [3]. Some purposes a solution with OAuth, so that the user does have to be redirected to every site while being logged out. But then the OpenID-provider has to be able to log the user out. Others talk about Andreas Solbergs's (FEIDE) solution with AJAX and iFrames using http-redirect [32].

Another mitigation to this problem, is to educate users on how to properly log themselves out of services or more functionality has to be implemented in browsers. The easiest solution would be to let the browser handle log-outs on behalf of the user if the user wishes to close the browser. Some extra intelligence has to be implemented into browsers that offer to store the users sessions. The browsers could either close the windows/tabs with active sessions and store the rest, or it could "click" on log-out on behalf of the user.

Although, if SLO is to be a part of OpenID, all the OPs and SPs would have to properly implement SLO for it to have the desired effect. Handling timeouts at different services can be a problem when implementing SLO [16]. By the looks of SLO's status today, an implementation has to be customized to offer good user experience [22].

Recent activity: With CardSpace users are able to see the last time they used a card and where it was used. And within OpenID, users have a view over the recent activity like when a successful or a unsuccessful login was made, whether we used password or InfoCard to log in, the services myopenid has approved to access my information and when we last signed out. There is also an overview over how many activities we had on each session.

Personal icon: myOpenID offers users to have a personal icon. This icon is to help users recognize their OP in case of e.g. an attempt on phishing. But it seems like a cookie or some sort is set to point to the picture of your choice. The reason is that; firstly, the icon is only available on the browser on the computer you are setting the icon. And secondly, the icon disappears when the browser history is cleared and the browser is restarted. A quite useless feature.

Trust

The initial trust is maybe the most important part. Users will have to trust in the system. A system where their credentials are given to a service in order to authenticate them to other services.

With the amount of information an OP is storing about the usage can be both good and bad. The user will have to trust that the OP does not sell this information to a third party. On the other hand, the information gives the user full control over their flow of identity information. Another problem is regarding the lack of support in the OpenID specification, which can have a negative effect given that the user finds out about it.

Consistent feedback: The interaction with CardSpace is consistent, there is not too much information to handle. Since the CardSpace is in focus (see figure 15) the user is force to pay attention to the given information. You can easily find out who the card is sent to and whether the card has been used before.

myOpenID provides sufficient information during authentication and consent. The user can easily see who they are dealing with. myOpenID is informing the user about the SP that initiated the authentication process and also the SP that require user's personal information.

Easy design: The CardSpace design is quite simple, the cards are in focus. If the user want some more information, a task-field is present (which is context-sensitive).

myOpenID's navigation is minimalistic, which is good for non-tech users, although the home screen is promoting their solutions and some SP-sites. This is probably their way of making money, which is understandable, but it could have an effect on someones trust.

Privacy

OpenID and Information Card technologies enables the ability for third-party identity providers to track and correlate user activities across any number of websites and services. Which is a huge privacy issue if the IdP decides to go rogue.

OpenID will allow providers to see who you access on the Web. Your OpenID provider could tell someone that you often visit online shopping sites, especially sites that sells shoes. This information could be valuable to shoe advertisers, they can then send you e-mails or show their an ad on your

Figure 19: myOpenID - Visited pages

OpenID-providers page.

The way OpenID ”handles” privacy is by involving the user when OP is sending information to SP on behalf of the user. The user is responsible for exposure of their personal information.

The safest way would be to run your own IdP, but that would mean that you can never sell/lose your domain. If someone else took over your domain, they could in theory take over your OpenID.

CardSpace only offers some privacy protection in specific use cases with self-issued cards. That would also mean that the SP would have to accept information cards and not log users visits.

Visited sites: myOpenID offers a list over all the services you have visited (See Figure 19), which is quite useful. You are able to get information about the persona your are using with the service, when you last were signed in, number of times myOpenID has approved their access to your information and you can choose to not let myOpenID auto approve for you.

Deletion of identity: When a user decides to delete her OpenID, the accounts at the service providers are still there. There is currently no standardized way to do this, you will have to visit every site and try to delete your accounts. Not every service provider offer a simple solution to delete your account. Ziki[20] is such a service that offers a link from the account detail page, while other services like Jyte[21], koornk[22] and ClearBits[23] does not mention this on their site at all. The procedure could be very technical and extensive, and not everybody has the patient to complete steps. This issue could be due to the lack of standardization of a deprovisioning protocol. SPML is such a protocol, but the problem with this protocol is the implementation. Developers has to customize in order to suite their needs [11].

A common option is to just forget the accounts, but this is not a very good choice, especially in terms of privacy. Although the user is responsible for giving consent, information about the user is still out there.

You could try to send an e-mail and hope that they could do this for you, but the problem would be that everyone could delete others account. Especially with social networks, where a lot of information is revealed to other users. An answer to a control question like: "What is your dogs name?" could easily be obtained. For example, if you want to delete you account at Facebook[24] you would have to send a special request to the administrators and then do not log in for two weeks. But if someone stole your account and deleted it you could restore it by again contacting the administrators. This part of the identity life cycle is maybe one of the most annoying part, especially for the user.

A mitigation to delete all your accounts associated with your OpenID-provider, is to have a function to delete your accounts with SP's based on your audit log. This is something that could be included in the OpenID specification.

The biggest privacy issue is when the account is hijacked. This would mean that the hijacker could get an overview of every site you have visited and how often. They could also log into every site and end up building a rather extensive profile about a person. But this is also something that could happen with e-mail accounts, so the risk is pretty much the same.

[20] www.ziki.com
[21] www.jyte.com
[22] www.koornk.com
[23] www.clear-bits.net
[24] www.facebook.com

Anonymity and pseudonymity: Although the OpenID specification has support for pseudonymity, myOpenID is not currently offering that feature.

Usability

Adoption: The users in the OpenID-CardSpace combination are required to either have Windows operating system with support for CardSpace or obtain a browser with CardSpace capability. Internet Explorer 7.0 and later has integrated support for CardSpace, while with other browsers users will have to install some kind of plugin first.

Users could feel that OpenID with CardSpace requires a lot from them. The idea is to move the users away from a familiar way to authenticate themselves. This could be one of the barriers users will have overcome before they switch to this solution.

Users will have to learn and get used a new way of signing in to services. Unless they have a computer with Windows Vista or later, they will have to install a browser with support for Information Cards or a stand-alone identity selector like DigitalMe[25] or Azigo[26]. This could be a reason for not using CardSpace. In terms of usability, in order for easy adoption of Information Cards would be to have browser with a pre-installed identity selector.

An easy way for users to adapt OpenID is to use an e-mail provider as your OpenID-provider. The largest e-mail providers like Google, Windows Live, Yahoo and AOL are OpenID-providers today. But none of them supports Information Cards as of today.

User-knowledge: Users will have to be aware of the components in the infrastructure. Without any knowledge about the infrastructure could result in situations where the user is not properly logged out or being a victim of phishing.

myOpenID does not mention anything about the log-out process, which is important information for the user. Users has to know what the behavior is when they wishes to log out of their OP's. Are they only logged out of their OP or does the OP help the users log out of all their ongoing services as well? Users might think that with SSO comes SLO.

Usage: The mental load of using CardSpace with OpenID is not so high, despite the number of steps required to get an user authenticated. Since

[25] http://code.bandit-project.org/trac/wiki/DigitalMe
[26] http://www.azigo.com/

OpenID offers SSO-capability, CardSpace is only invoked at initial login and when the OP's session times out. During authentication, users only have to type in their OP's URL and the rest of the process can be done by clicking on the mouse. When the user is already logged in, the following logins to other SPs will just require the user to type in the OP's URL.

Some users could find the invocation of the CardSpace annoying, because of its resembles to Windows User Account Control (UAC). When UAC was first introduced in Windows Vista, a lot of people thought it was irritating. So they turned it of. That could be a problem when users are introduced to CardSpace. This issue affects usage and adoption.

9.3 Discussion

This discussion will consider whether Information Card can be a suitable substitute for regular username and password. Note that this experience is only with Windows CardSpace.

The metaphor for information cards is understandable to many individuals, and most people will be able to understand their function, without the necessity of understanding its technical complexity.

CardSpace eases the authentication process and enhances the user experience. With that being said, CardSpace do require more steps to get authenticated than just using an username and password. If we start from a service provider; the user only has to type in myopenid.org, be redirected to the login screen, choose to log in with an InfoCard, wait for CardSpace to appear, choose a card, review the card, send the card and get access. Users do not have to type in their username or password, and CardSpace even suggest the right card if the user has previous visited the OP. All of this can be done by clicking, which can be useful on portable devices and touchscreens.

With CardSpace, users are getting more than just authentication help. Information cards are more intuitive and convenient, and no username and passwords are sent during authentication (more secure). CardSpace can detect and warn us if we are sent to another URL than the last one we used. But this is not something the user is told unless the user encounters it or she reads about it.

myOpenID offers the ability to remove the use of passwords if the user has associated the account with an information card. With the current state of information cards today, users would have to export their cards to other devices manually. Meaning that there is no easy way to synchronize your

cards between different platforms or devices. A very nice feature offered by myOpenID though, is to let users manage multiple information cards. This means that the user can use CardSpace on different locations to log in. But this also means that the user would have to secure several systems and make sure that no one gets access to her cards. A PIN in CardSpace could protect the cards, but the PIN's are there to make sure that no one sends your cards. Others can still see which services you are using by looking at the cards name or logo.

But in order to make the adaption of identity selectors less of a burden, identity selectors would have to be included in web browsers. There are solutions for identity selectors like CardSpace on other platforms, but the interface is not the same and the experience is full of bugs. Identity providers and service providers could be somewhat of a hassle for normal users. The thought of having to log onto a service to then log on to other services can be a startup bump.

10 The future of federated identity management

This section is more of a discussion part, where we will address some of the thoughts and visions for federation in different areas in the future.

The federated identity model is not a model that is suitable and scalable for online internet users that rapidly moves from sites to sites. This is due to the disadvantages in federated identity, such as the high technical and legal complexities, and the fact that it is unimaginable for all SPs to federate. Federations makes much more sense inside a enterprise or a government-based service. It is in this environment where a central force may enforce such a large and complex change. This is also an environment dependent on high trust requirements from each entity. The information in these systems are highly sensitive, whether there are company secrets or employee records. The web services in a federation has to be related to each other to make any sense.

The user-centric model seems to be moving in the right direction to meet users requirements in terms of usability. But until we have real client-side SSO, the privacy issues will still be there (IdP). We have examples of governments that are ready to adapt this model, but the meaning is not to replace the federated model but to complement it. As of today, user-centric identity is mainly intended for low risk services. Security issues, security requirements and business motivations has to be properly addressed before the user-centric model can really compete with the federated. In areas where there may not be natural with a federated model, like online services and communities, an user-centric model can give these areas more value and more activity.

We have mentioned aspects like trust, usability, frameworks and data quality assessment. The common denominator between these aspects is that they not necessarily can be solved with the help of technology. Economic and legal issues has to be laid out and conquered before some kind of federation can take place, or not. The current and future development of federated identity management has and will be different for areas like; eCommerce, eGovernment, enterprise and online communities. We will have a look in these areas and look at the driving forces behind federated identity and suggest some use cases. At the end of this section, we want to see whether user-centric identity have the needed interest and if the market have the motivation to convert or combine it with federation. We want to focus our discussion on what federation can or can not offer the users in the future.

10.1 Online services

By online services, I refer to services like web shops, auction sites and subscription sites (like news sites and magazines).

It is not very likely that online services are going to form a federation without some beneficial agendas or if their businesses are related (like insurance and banks). The main reason is that a federation require a very tight trust relationship and complex business agreements. As long as users need access to services across several federations and then some, the need of having several accounts and thus several username and passwords is still there. Password fatigue is just temporarily patched and it is not likely to see all your services under one federation.

But one aspect of online services where federations could benefit the users would be that the users information could be changed at only one place. Every time an user is going to buy something, her address could be pulled from a place with updated information. Lets say that a person subscribes to different magazines that are mailed to her home every month. If the person were to move somewhere else, she would have to log onto every magazine website and change her address. Services would have to either connect with each other or to a service that offered the possibility to subscribe to some sort of personal database. This problem could also be mitigated by accepting e.g. an updated personal information card.

User-centric?

Online services could drop the responsibility of maintaining an user database. They do not have to worry about privacy, provisioning and security of users personal information. Users can have more control over their identities with the help of their identity wallet, and also have more control of the services they have visited. This could result in users returns to a service because they no longer has the problem of forgetting a password or the service.

I also think that we will get increased activity on sites that offers attribute exchange during signup. This is very inviting feature that service providers can benefit greatly from. Especially, for users that turn around and leave when they have to register to a service before they can use it. This could mean a bigger flow of users, visiting and using the site. Or at least check it out.

Another useful feature is for example student discount at certain eCommerce sites. Instead of making a new account at the web store and sending in your student card for approval, which can take days or weeks, the site could support e.g. OpenID to make sure that an user is a student

of a certain university. The parties would save time and the store could potentially get more customers to the site.

10.2 eGovernment

eGovernment in this thesis is referred to as the service that government offers to their citizens.

This is an area where we will see more federated activity. eGovernments is all about collecting available services and presenting them under one interface. There are discussion about medical data, and how to offer health services to the citizens through eGovernment. Frameworks and policies has to be followed in order to access data government-kept data. Since the data of a citizen is very sensitive data, the government has to be sure that every party that joins the federation is trustworthy.

We are intentionally looking away from the extremely complex legal implications of some of the following future scenarios and trying look at them with an idealistic mind.

A possible area of cooperation between government federation in different countries is visa applications. The process could be made automatic and citizens that want to travel to another country would be able to get an answer within seconds. It would of course make it easier for the police to track someones habits and enable a "big-brother" world.

Another area where a collaboration could the benefit the citizen is health cards that users can carry around when traveling abroad. If a person had to go to a hospital in another country, she could swipe her card and informations like allergies, previous illness could be made available for the doctors for a period of time. Germany had a similar project within Germany, called Health Insurance Card, but this project is currently set on hold due to security and politics [9].

User-centric?

Governments are talking about eGovernment 2.0, where publicly forums are created where citizens can log on with a government-approved IdP. They will be able to communicate with the government in an anonymous way (or at least pseudonymous). The reason the IdP has to be government-approved could be that only citizens of that country or state has access to the forum. The government also want the IdP to be certified in someway to meet their security requirements. The government in Japan has started to accept OpenID to a website called IdeaBox, a place where people can propose, dis-

cuss and vote on policies. But they are only accepting OpenID from mixi, Yahoo! Japan, Livedoor and Google [29].

There is not very likely, in the near future, that a model like the user-centric will get access to high risk applications.

10.3 Enterprise

Enterprises in this thesis is referred to as services that are offered to its employees within an organization.

This is also an area where we will see more federated activity. Enterprises uses federations to achieve more productivity and seamless user experience for their employees. They are always trying to connect all their systems to make it as convenient as possible, but there are still a lot of technical and legal issues that has to be sorted out before the implementation can start.

From my point of view, it is very hard to find new use cases for enterprises when it comes to federation, due to their closed environment and different domains. But enterprises could connect their systems together by connecting every offices globally, to gather information in order to ease the transition of employees moving abroad (but within the enterprise).

These use cases has to be specific to each domain and that is way a general assumption is hard to find.

User-centric?

An user-centric approach could be an easy way to give customers or employees temporarily access to your systems, in the case where a federation is highly unlikely or too complex. Employees would like access to both their clients systems, as well as their own. The suggested scenario would be to let the customer be authenticate at her system, which would send an assertion to your system where you would be able to authorize her and give her access. All of this without the need of creating a new user account. She can do her work by just logging in once. The work of updating personal information, provisioning and revocation would be handled by her home company. Compared to the unnecessary maintenance if the user were to create an account at her client. This is just in theory.

Another useful feature is to have employees participate in for example public forums, where users would get to discuss with IT-professionals.

A third use case is employee discount at different web shops. The shops would immediately know that the URL was from someone working at that company, because they knew that only employees would get that URL. To secure this feature, the web shops could store the companies certificates to ensure that it is not a similar URL.

There has been some talk about SUN enabling OpenID for all of their employees and that only SUN employees would get an "*.openid.sun.com" identity. But there is not much talk about this after their release in 2007. The agenda behind this adaption of OpenID was to explore the two latter use cases.

10.4 Online communities

Online communities in this thesis is referred to as social networks, blogs and forums.

It is not likely that online communities like Facebook[27] and MySpace[28] are going to form a federation. They all have their valuable user databases and their business models. The complexities behind federated identity so high that most online communities would not consider it, plus that the motivation is not presented.

Although the online communities is not likely to federated, the users would want to "federate" themselves. There is a demand for a platform where you can gather all your online identities and build some kind of a social reputation. Users will probably not mix their work-life community with their social community. But whenever a new social network arises, users would like to connect with their friends on the new network without having to find everybody and reconnect again. So an user-centric approach with the ability to bring your connections with you to different sites could be a way to address this demand.

User-centric?

Google and MySpace has developed a set of APIs, called OpenSocial[29], in order to create a service that can support multiple social networks. This could be the next step in collecting all your social identities with your friends, pictures and information under one interface/portal.

[27] www.facebook.com
[28] www.myspace.com
[29] http://code.google.com/intl/no/apis/opensocial/

By adapting a user-centric approach, the activity on online communities could grow and make it easier for ”lazy” user to register to the service and use it. Since there are someone/something handling their login credentials, the possibility that users stops using a service because they forgot their username and password goes down. Users will have more control over their identities and the chance of them forgetting a service due to inactive use would also be reduced. But the most important downside for a SP is that they cannot collect user information in the same way they did under a silo-based model.

Another exciting area is user-initiated content aggregation. Imagine that you have a service that host your pictures, a service that host your music and another service that host your documents. There are services that allows users to collect content from these kind of sites and presenting them under one interface. OAuth can be used to allow users to share their private resources stored on one site with another site without having to hand out their username and password.

11 Conclusion

We have in this thesis looked at some of the future possibilities regarding federated identity management. We will try to answer the research questions in our introduction.

11.1 The future of federation regarding online services, eGovernment, enterprises and online communities.

I have tried to suggest some idealistic, but still feasible use cases for online services, eGovernment, enterprises and online communities.

For most of the areas, it seems like the agenda is about making the world smaller. It is about gathering information or making it available no matter where you are.

Online communities is probably the only area where we will not see any activity regarding federation. The complexities behind federated identity so high that most online communities would not consider it, plus that the motivation is not presented.

11.2 How will some of those areas benefit from adapting user-centric technologies?

The benefits are very different from area to area, but it seems like some of most of them; eGovernment, enterprises and online services, can get more value in socializing with customers/citizens. Online services can get more user to adapt their services. Governments can reach out to their citizens and communicate with them. Enterprises can benefit from an user-centric approach in an ad-hoc environment for a period of time, like consultant work for another company. Some of them are already under development while others are only in the planning stage.

The area with the activity of user-centric development is online communities and low-risk online services, which was the intention for OpenID in the beginning. We will see more of social reputation services, where those services or networks would pretty much be like the networks in real life.

We are going to see that all of the mentioned areas will benefit from adapting user-centric technologies. It will start in mostly low-risk services, and once the most problematic issues is properly addressed, the higher risk services will emerge.

11.3 Password fatigue: How is federated and user-centric identity really dealing with this problem?

Federated:

Federated identity management aims at collecting all of your identities within the federation and letting users have access to all of them by authenticating once.

Problems with federations regarding the password fatigue problem is first of all the high trust requirements. This means that it is not likely that unrelated services will form a federation. Small services with no need, no resources or the technical requirements to go through with it. As mentioned before, the areas where federations will benefit their users is in governments and enterprises. They are dealing with services of higher risk and their assets has to be properly protected. That is why their motivation is stronger and enough resources and the competence to start this kind of project.

Federated identity solves a lot of problems within an area. In terms of employees getting more access with only one set of credentials, federated identity is a great to enable that. But for users on the internet that uses a big variety of services, both related and unrelated ones, can not get the seamless experience federation can offer within a company. Federation do reduces the number of username and password needed, but to a certain point. We cannot expect that every services on the internet is going to gather under one federation.

It is maybe wrong to say that federated identity management is not solving the password fatigue problem, because they do solve it. Maybe not for regular internet users, but federation is certainly doing its job within organizations for employees.

User-centric:

My evaluation showed that there is a lot of potential if user-centric technologies get more adaption. CardSpace or Information Cards can make authentication more intuitive and more secure if they are implemented properly. Higgins are working in a browser-integrated client, called Active Client 2.0[30], which supports Information Cards, OpenID and username and passwords (CardSpace is also going to support all of them as well). So the users that are skeptical about Microsoft products can hopefully in the future adapt Higgins Active Client 2.0 or something similar. As for now, CardSpace is only checking servers URL against phishing. A better solution could be to

[30] http://wiki.eclipse.org/Active_Client_Overview

store servers certificates instead and check if they match next time the user wishes to use the service.

OpenID offers SSO, access to all their services by authenticate themselves once and that they give access to a large number of services. Features that are great for users that seeks to make authentication less annoying and better user experience. The user experience is very pleasant together with CardSpace. This solution is a strong candidate to replace username and password, at least for most internet users.

I really believe that Information cards be a candidate in replacing username and password, because of its intuitive metaphor and function. Especially after my evaluation.

But OpenID has room for improvements beyond the phishing problems. CardSpace is a easy way to mitigate phishing, but new more advanced problems could arise on that front. OpenID should look into supporting SLO, especially since IdPs are not telling user to log out of their active services. Better user education could make users more aware of threats, but it is hard to educate every user on the internet.

User-centric technologies could be combined with federations to allow users to support several federations by e.g. saving the credentials to a federation on a device or an identity selector [20].

11.4 Some last thoughts:

Federation is not an improvement that suits everybody, that is why the user-centric could be a step further in the identity management chain to create more value for the users.

Before user-centric identity can become a serious alternative to federation or cooperation, better business use cases has to be created. There has to be more beneficial use cases where e.g. transaction fees or an economic infrastructure can address the needed income and interest for IdPs. Once these road bumps has been properly addressed, companies would invest more resources into making a secure infrastructure where users can move freely with their e.g. identity wallets. Without these use cases, an IdP will not have the proper motivation, at least not in the commercial world.

IdPs have of course a business model by tracking their users usage and utilize this information in some way. They can use this information present ads that reflect users habits, in the same way Google's mail service does. No information is shared with others, but Google is matching your information

with different ads. Everything is done automatically. Some users would be very critical about this. And to stop the ads, the IdPs could offer a pay-service with higher security and no-ads. This is of course a scenario where there users are ”aware” of the aggregation of information. The worst case scenario would be to sell the information to someone else, so that they can generate a powerful profile about you.

12 Future work

Based on the findings in my evaluation and some general thoughts regarding the future, these topics are worth discussing further.

Using Information Cards for non-browsing applications:

An interesting scenario would be to connect your portable device to an operating system in order to get authenticated. You could choose a card from your device and send it to the operating system. Pretty much the same way we are using finger scanners on some laptops today.

More reliable applications with OpenID:

One possibility to ensure that only ”trusted” IdP could be used on a site would be to make a global black-list of some kind. Another possibility would be to inherit a hierarchical model, where some IdPs would be more trustworthy than others. This could be similar to Certificate Authorities in the X.509 model.

The problem with both these suggestions is that it goes against OpenIDs purpose of being lightweight and open. Another problem is, who is going to decide whether an IdP is going to be banned or get an higher status than other IdPs? Some governments are already doing this.

Problems with logout:

Users tend to close their browser to logout from services. But there are a lot of scenarios where the browser do not exit properly, leaving the session open. It could be that the users have another window open, do not close the application or have a ongoing download. The behavior of different browsers and operating systems forces the users to act differently when they are moving between e.g. work and home. Imagine a scenario when an employee is working with three services and she chooses to use three browser windows, instead of tabs. It is lunch time and the employee decides to log out, she clicks on the logout-button in her current window. The service tells her that she is logged out from all of her services. She leaves her desktop and forgets to close down the two other windows. This means that the remaining windows could contain some sensitive information is still visible. The problem would be more severe if SLO were not implemented. That would mean that the two remaining windows were still logged on, giving another user full access. This problem could potentially be solved by the browsers or better user education.

Deprovisioning:

Poor support for deprovisioning can have consequences on privacy. By having information about you floating around the web is not something most people would want, if they could do something about it.

There should be done more work on standardizing provisioning and deprovisioning [11]. Making it easier for services to offer deprovisioning and making sure that they offer it. This could be a very irritating factor for users, especially when they do not find an easy way to delete their accounts.

Another issue with deprovisioning is the case where you want to delete all your accounts created with OpenID. There is not a way to automatically delete all your accounts from the IdP, unless you go over your history of visited sites, visit them and trying to find out how to delete your account.

13 References

References

[1] Uninett ABC. Temahefte: Datavask og rutiner - beste praksis. http://www.feide.no/sites/feide.no/files/documents/temahefte_datavask.pdf, 2007. [Online; accessed 20-Mar-2010].

[2] J. Altmann and R. Sampath. Unique: A user-centric framework for network identity management. In *Network Operations and Management Symposium, 2006. NOMS 2006. 10th IEEE/IFIP*, pages 495–506, April 2006.

[3] Nate Klingenstein Andrew Arnott, Joost van Dijk. http://lists.openid.net/pipermail/openid-general/2009-september/019186.html. http://rnd.feide.no/content/how-create-a-fancy-iframe-demo-a-z, 2009. [Online; accessed 22-Apr-2010].

[4] Abhilasha Bhargav-Spantzel, Jan Camenisch, Thomas Gross, and Dieter Sommer. User centricity: A taxonomy and open issues. *J. Comput. Secur.*, 15(5):493–527, 2007.

[5] Jim Bidzos. Keynote: Impact of design on trust. http://blogs.verisign.com/web-user-experience/2010/03/impact-of-design-on-trust.php, 2010. [Online; accessed 14-April-2010].

[6] Cathy Soohoo David Danielson Leslie Marable Julianne Stanford B.J. Fogg, Ph.D. and Ellen R. Tauber. How do people evaluate a web site's credibility? http://www.consumerwebwatch.org/pdfs/stanfordPTL.pdf, 2002. [Online; accessed 5-Jan-2010].

[7] Kim Cameron. New prototype could really help openid. http://www.identityblog.com/?p=1070, 2009. [Online; accessed 16-Mar-2010].

[8] Michael Cobb. Preventing password fatigue with single sign-on (sso) authentication. http://searchsecurity.techtarget.co.uk/tip/0,289483,sid180_gci1378494,00.html, 2010. [Online; accessed 3-Mar-2010].

[9] Tony Collins. Germany shelves č1.5bn e-health card scheme. http://www.computerweekly.com/blogs/tony_collins/2010/01/germany-shelves-15bn-e-health.html, 2010. [Online; accessed 21-April-2010].

[10] Lorrie Cranor and Simson Garfinkel. *Security and Usability*. O'Reilly Media, Inc., 2005.

[11] Mark Diodati. Spml is on life support... http://identityblog.burtongroup.com/bgidps/2010/02/spml-is-on-life-support-.html, 2010. [Online; accessed 16-Apr-2010].

[12] Joseph C Dolson. What is web usability? http://www.joedolson.com/what-is-web-usability.php. [Online; accessed 13-Mar-2010].

[13] Crc For Enterprise, Audun Jøsang, and Simon Pope. Auscert conference 2005. In *in Asia Pacific Information Technology Security Conference, AusCERT2005, Austrailia*, pages 77–89, 2005.

[14] Credential Federal Identity and Access Management (ICAM). Identity metasystem interoperability 1.0 profile. http://www.idmanagement.gov/documents/ICAM_IMI_10_Profile.pdf, 2009. [Online; accessed 25-Mar-2010].

[15] Credential Federal Identity and Access Management (ICAM). Openid 2.0 profile. http://www.idmanagement.gov/documents/ICAM_OpenID20Profile.pdf, 2009. [Online; accessed 25-Mar-2010].

[16] FEIDE. Single sign on - single log out. http://docs.feide.no/fs-0034-1.0-en.html, Unknown. [Online; accessed 26-April-2010].

[17] Google. Google apps. http://www.opengroup.org/projects/idm/uploads/40/9784/idm_wp.pdf, 2010. [Online; accessed 20-Mar-2010].

[18] Graham Hayday. Security nightmare: How do you maintain 21 different passwords? http://www.silicon.com/technology/security/2002/12/11/security-nightmare-how-do-you-maintain-21-different-passwords-11036760/, 2002. [Online; accessed 13-Mar-2010].

[19] Ashish Jain. Apache authentication module for cardspace. http://itickr.com/?p=56, February 2007. [Online; accessed 3-Feb-2010].

[20] Audun Josang. Lecture 7: Identity and access management. http://www.uio.no/studier/emner/matnat/ifi/INF3510/v10/learningdocs/INF3510-2010-L07.pdf, March 2010. [Online; accessed 23-Mar-2010].

[21] Audun Jøsang, Muhammed Al Zomai, and Suriadi Suriadi. Usability and privacy in identity management architectures. In *ACSW '07: Proceedings of the fifth Australasian symposium on ACSW frontiers*, pages 143–152, Darlinghurst, Australia, Australia, 2007. Australian Computer Society, Inc.

[22] Andreas Åkre Solberg. My thoughts about slo. http://rnd.feide.no/content/my-thoughts-about-slo, 2009. [Online; accessed 14-April-2010].

[23] Eve Maler and Drummond Reed. The venn of identity: Options and issues in federated identity management. *IEEE Security and Privacy*, 6:16–23, 2008.

[24] Norway Ministry of Government Administration Affairs (FAD). Rammeverk for autentisering og uavviselighet i elektronisk kommunikasjon med og i offentlig sektor. http://www.regjeringen.no/upload/FAD/Vedlegg/IKT-politikk/eID_rammeverk_trykk.pdf, 2008. [Online; accessed 12-Dec-2009].

[25] Australian Department of Finance and Deregulation. National e-authentication framework. http://www.finance.gov.au/e-government/security-and-authentication/docs/NeAF-framework.pdf, 2009. [Online; accessed 12-Dec-2009].

[26] Department of Finance and Deregulation (Finance). National e-authentication framework. http://www.finance.gov.au/e-government/security-and-authentication/docs/NeAF-framework.pdf, January 2009. [Online; accessed 05-Dec-2009].

[27] A. Pfitzmann and M. Hansen. Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management - consolidated proposal for terminology v0.31. http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.31.pdf, 2008. [Online; accessed 22-Feb-2010].

[28] Anna Pickard. Are you suffering from password pressure? http://www.guardian.co.uk/technology/2008/jan/17/security.banks, 2008. [Online; accessed 13-Mar-2010].

[29] Nat Sakimura. Government of japan started accepting openid. http://openid.net/2010/03/09/government-of-japan-started-accepting-openid/, 2010. [Online; accessed 25-April-2010].

[30] Sarah D. Scalet. The truth about federated identity management. http://www.csoonline.com/article/221034/The_Truth_About_Federated_Identity_Management, 2006. [Online; accessed 3-Mar-2010].

[31] Thomas J. Smedinghoff. Legal obstacles delaying federated identity management. http://www.cio.com/article/178001/Legal_Obstacles_Delaying_Federated_Identity_Management?page=2&taxonomyId=3089, 2008. [Online; accessed 2-Feb-2010].

[32] Andreas Solberg. How to create a fancy iframe logout demo - from a to z. http://rnd.feide.no/content/

how-create-a-fancy-iframe-demo-a-z, 2008. [Online; accessed 22-Apr-2010].

[33] The MIT Kerberos Team. Kerberos: The network authentication protocol. http://web.mit.edu/kerberos/. [Online; accessed 16-Feb-2010].

[34] Uninett. Moria web authentication service. http://moria.sourceforge.net/index.html, 2006. [Online; accessed 25-Jan-2010].

[35] UNINETT. Feide integration guide. http://www.feide.no/sites/feide.no/files/documents/Feide_integration_guide.pdf, 2009. [Online; accessed 25-Jan-2010].